Topology Reconstruction via Path Recording in Secure MANET

Danai Chasaki, Y. Sinan Hanay and Tilman Wolf
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, USA
Email: {dchasaki,hanay,wolf}@ecs.umass.edu

***Abstract*—The exchange of topology information is a potential attack target in mobile ad-hoc networks. To provide an intrinsic security mechanism, it is possible to validate topology advertisements in the control plane against records of the path taken by transmission in the data plane. In this context, we provide a discussion of different path recording mechanisms. We evaluate their performance in terms of packet overhead and reconstruction complexity.**

I. Introduction

Mobile Ad-Hoc Networks (MANETs) are essential communication infrastructure in the Department of Defense's vision Network Centric Warfare [1]. In particular, as part of the Global Information Grid (GIG), MANETs can serve in Warfighter Information Network-Tactical (Win-T) to provide command, control, communications, computers, intelligence, surveillance and reconnaissance (C4ISR) support [2].

A particular interest in this space is the design of MANETs with intrinsic information assurance properties. The key aspect of intrinsic assurance is that security properties are achieved by the inherent design of systems and network protocols rather than by added features on top of insecure networks. Such a design approach requires a fundamentally different approach of networking than is common. Conventional protocol designs often assume a cooperative environment, where other network nodes participate in the protocol implementation with no malicious intent. Intrinsic assurance requires that such assumptions not be made, and instead all nodes be considered as potentially malicious.

With scenarios where other network nodes cannot be trusted by default, it is important to identify what information in a protocol exchange can be trusted. In our work, we focus on network topology information (i.e., which node can communicate with which other node). Due to the dynamic nature of MANETs this information changes during the operation of the network and thus cannot be predetermined. Typically, there are two sources of information from which topology information can be obtained:

- Topology information via control plane: Routing message exchanges between nodes (e.g., link state updates) provides a complete view of the network.
- Topology information via data plane: Data transmission along paths in the MANET can only travel along valid links and thus implicitly reflect topology information.

To validate information that is obtained from one source it can be verified against information that was obtained from the other source. Such validation allows the identification of discrepancies, which is the first step in identifying and isolating malicious nodes and thus achieving a secure communication environment.

In this paper, we address the problem of how record data path topology information in an efficient manner. Recording and exchanging control information in the control plane has been studied extensively in the context of routing protocols, but the data plane has received little or no attention. We explore several mechanisms for efficiently recording this information and exchange it securely between nodes. The specific contributions of this paper are:

- A discussion of different path recording techniques that are deterministic or probabilistic in nature.
- A quantitative evaluation of the performance tradeoffs between these techniques in terms of space and computational requirements.
- A discussion on how to provide security in path recording mechanisms.

The remainder of the paper is organized as follows. Section II introduces related work. The general process of topology reconstruction from path records in illustrated in Section III. Specific path record data structures are introduced in Section IV. The packet overhead and reconstruction computation is quantified in Section V. Section VI summarizes and concludes this paper.

II. Related Work

Intrinsic assurance in network designs has been proposed in the form of "off-by-default" network architectures [3]. Instead of allowing any node to connect to any other node as proposed in the original Internet architecture, nodes need to obtain explicit permission. Examples of such an architecture are capabilities-based networks, which also have been proposed in the context of military networks [4].

Extracting topology information from a network has been discussed in different domains. A commonly used method is

This material is based upon work under a subcontract #069153 issued by BAE Systems National Security Solutions, Inc. and supported by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare System Center (SPAWARSYSCEN), San Diego under Contract No. N66001-08-C-2013.
978-1-4244-2677-5/08/$25.00 ©2008 IEEE

network tomography or other inferential network monitoring and is based on end-to-end traffic monitoring to uncover internal network characteristics. Topology reconstruction techniques based on end-to-end delays of multicast traffic are proposed in [5] to infer the multicast tree.

Tian and Shen also propose an algorithm which determines the topology of a network based on end-to-end measurements in [6]. Probe packets are sent from some sources towards multiple destinations, and each pair of nodes keeps track of the packets received. The nodes on which multiple links converge share the information about packet loss or delay of multiple links. Thus, the correlation of the received information can be compared and through statistical methods the whole network tree can be reconstructed.

The use of mobile agents has also been considered as a solution to the topology discovery problem. A mobile agent is a controllable program that can move inside a network. For topology discovery, several mobile agents traverse the network to collect topology information and transmit this information back to a centralized management station [7].

Our approach to topology reconstruction differs to these inference approaches insofar that we explicitly record the path that a packet takes through the network. This requires a change in the packet header and packet forwarding routine. In the context of secure MANETs, this is a reasonable assumption since their security design requires many other additional changes.

III. Topology Reconstruction via Path Recording

We begin our discussion with a general overview of how topology reconstruction fits into the context of security in MANETs.

A. Overview

The main concept of how path recording and topology reconstruction are related to security in MANETs is illustrated in Figure 1. In the control plane, routing information is exchanged between nodes. The mobile wireless nature or MANETs implies that practically any node could be connected to any other node at some time. It is therefore difficult to make inferences on the correctness of topology information that is gained via routing information exchanges. A malicious node could advertise connectivity to any other node. This action could be the basis of a black hole attack where traffic is attracted through routing and then not forwarded in the data plane. (In contrast, consider a wired network: Due to the fixed topology, routers can only advertise connectivity to their neighboring nodes. Thus, a malicious advertisement of false connectivity could be detected immediately.)

To address this problem in MANETs, we present a path recording mechanism that allows the reconstruction of the topology from the point of view of the control plane (see Figure 1). By recording the nodes and links that a packet traverses, it is possible to identify what connectivity really exists in the network. This information can then be used to reconstruct a topology that can be compared to that obtained

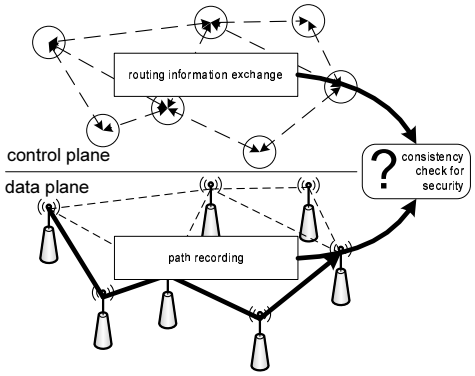


Fig. 1. Validation of Control Plane Information with Path Recording in the Data Plane.

from the control plane. Validation of both topologies can be a first step towards identifying nodes that do not behave as they are supposed to.

In this paper, we focus on how to record that path that a packet takes and how to reconstruct this information at the receiving end. We explore different data structures for path recording and evaluate their space requirements (in the packet header) and determined the computational cost for reconstructing the topology. In this context, we make the following assumptions:

- We focus solely on recording the path and reconstructing this information at the receiver. We do not consider the issue of how to react to discrepancies between control plane and data plane information.
- We assume that partial topology information is sufficient for a node to identify some discrepancies. To obtain a complete view of the network, traffic would need to traverse all possible links and end up at a single node. Since this is unlikely to occur, we focus on reconstructing the path from source to destination from one end-system to another. It is implied that by repeating this process, additional paths from different nodes can increase the amount of topological information available to the receiver.
- We assume that the network is stable at the time scales considered for path recording. While MANETs are inherently dynamic, the duration of reconstructing a path is short and thus short-term stability can be assumed.
- In contrast to topology inferencing techniques used in related work, we are not limited to observing the network only from the point of view of an end-system. We assume we are able to change the data path operations of the network (i.e., by introducing additional header fields and computation on forwarding nodes).
- While we refer throughout the paper to recording a “path,” we understand that this work can also be applied to a network which employs network coding [8]. In the case of network coding, a mixture of packets is transmitted and the path becomes a “subgraph” of the network topology.

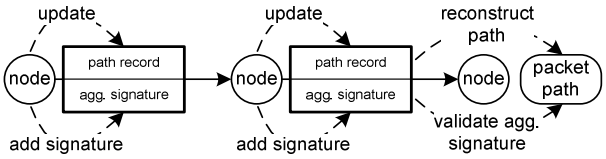


Fig. 2. Protection of Path Recording Data Structure with Aggregate Signature.

B. Security Issues with Path Recording

While path recording is intended to help improve security in MANETs, it is also important to consider how an attacker may take advantage of such a feature for malicious purposes. One of the simplest approaches to abusing path recording is to introduce incorrect information in the packet header data structure. To avoid this attack, the path record data structure is protected by a cryptographic signature as shown in Figure 2. When a node adds its topology information, it also adds its signature. To avoid problem associated with variable-length headers, we do not simply chain the signatures, but use a fixed-length aggregate signature as proposed by Boneh et al. in [9]. When the packet has reached its destination, the node can extract the path record and reconstruct the packet's path. Using the identifiers of all nodes along the path, the receiver can then verify that the aggregate signature corresponds exactly to those nodes. This indicates that the path record has not been tampered with.

If a malicious node attempts to tamper with the path record function, then it can be detected:

- Tampering with path record: If the path record is modified without correctly adjusting the aggregate signature, then a mismatch can be detected.
- Tampering with signature: If the aggregate signature is modified, then it does not correspond to the nodes represented in the path record, which can be detected.
- Omission of record: A node can chose to not record its information in the path record or signature. In such a case, the node will not show up in the topology recorded by the receiver. This behavior can be detected by comparison with routing information.

These scenarios show that an attacker cannot introduce incorrect information into the system. In the worst case, an attacker can deny the path recording feature by constantly invalidating the header fields.

IV. Path Recording Data Structures

We now turn to the question of how path information can be recorded efficiently. The main concerns in terms of efficiency are the following two quantitative performance metrics:

- Data structure size: The amount of space needed for storing the path record data structure determines the size of the header field necessary for recording.
- Reconstruction time: The number of computations necessary for reconstructing the packet path from the path record determines the computational overhead on the receiver.

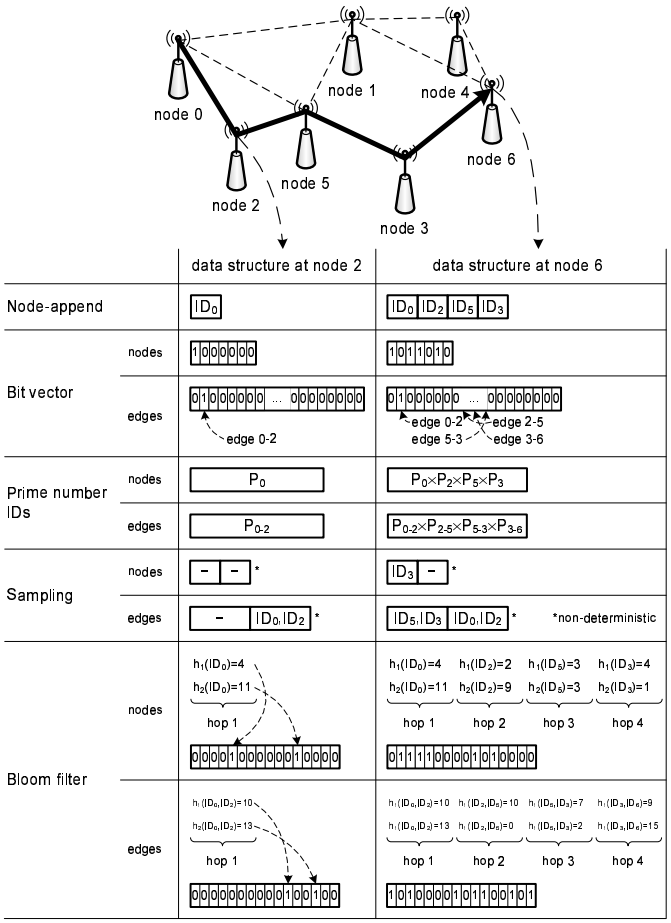


Fig. 3. Path Recording Methods.

We explore these metrics in more detail in Section V. First, we describe different data structures that can be used for path recording.

There are two fundamentally different approaches to recording path information: deterministic path recording and probabilistic path recording. Deterministic approaches reconstruct the path by keeping all the path information in a packet. The entire path can be reconstructed without error from a single packet. Probabilistic approaches record partial path information in a packet. Reconstruction may be inaccurate or require multiple packets. Typically, probabilistic methods use smaller data structure and thus create less overhead in the packet header. In our work, Methods I–III are deterministic and Methods IV and V are probabilistic.

Depending on the use of path recording, it may be necessary to record the *nodes* of the network that were traversed or the *edges* (i.e., links) that were traversed. Where applicable, we present algorithms for both approaches. All methods are illustrated in Figure 3.

A. Method I: Node Append

In this straightforward method, each node on the path appends its ID to a variable length header field of the packet.

The procedure is as follows:

- Data Structure: An initially empty, variable-length sequence of node IDs.
- Recording Operation: Each node adds its node ID at the end of the existing sequence.
- Path Reconstruction: The receiver extracts the sequences of node IDs to obtain the ordered list of traversed nodes and edges.

In the example shown in Figure 3, the packet starts traversing the path at node 0, and arrives at node 6 passing through node 2, 5, and 3. When node 2 receives the packet, the value in the header field is ID_0. Node 2 appends its ID (i.e., ID_2) next to ID_0. The same procedure is followed in subsequent nodes.

This method can only record a single path, not a subgraph that represents a network coding mixture since edges are inferred from the linear sequence of nodes.

B. Method II: Bit Vector

This approach uses a bit vector in which each bit position represents a node. The procedure is as follows:

- Data Structure: A fixed-size bit vector with a position for each node/edge initialized to all zeros.
- Recording Operation: Each node/edge sets the bit to 1 that corresponds to its ID in the bit vector.
- Path Reconstruction: By checking which bits are set in the final bit vector, an unordered list of traversed nodes/edges is obtained.

This method can provide a path or a network coding subgraph if edges are recorded. If nodes are used, no ordering information is maintained.

C. Method III: Prime Number IDs

In this method, each node or edge is assigned a prime number as an ID. The path record carries the product of all visited nodes/edges. The procedure is as follows:

- Data Structure: A (sufficiently large) integer initialized to 1.
- Recording Operation: Each node multiplies the integer with its prime ID and stores the result as the new integer.
- Path Reconstruction: The received integer is factorized to obtain the IDs of nodes/edges involved.

An example of the process is illustrated in Figure 3. This method can record an arbitrary subgraph and thus is suitable for network coding.

D. Method IV: Sampling

As one of the probabilistic methods, sampling records some of the nodes/edges along the path of a packet. With multiple packets collecting different samples, the path can eventually be reconstructed. The procedure is as follows:

- Data Structure: A fixed-length data structure with one or more places to record node/edge IDs.
- Recording Operation: With a certain probability, a node stores its ID (or the ID of the incoming edge) in the data structure.
- Path Reconstruction: When receiving a packet, the recorded nodes/edges are stored. As more and more samples arrive, an unordered list of nodes or an ordered list of edges can be obtained.

A key parameter for this method is the number of samples taken in a packet. In the example in Figure 3, two samples are used. Sampling edges is suitable for recoding network coding subgraphs.

Various improvements to this method have been published in literature. It is possible to compress path information by hashing [10] and to probabilistically select which nodes/edges record their ID.

E. Method V: Bloom Filter

A Bloom filter is a data structure that can efficiently store membership information [11]. To add an element to a Bloom filter, several hashes of the element are computed. The bits at the bit positions provided by the hash functions are set to 1. To test for elements in the Bloom filter, the hashes are computed and it is checked if the corresponding bits are set. When testing for membership information, it is possible to obtain false positive answers. The procedure for path recording is as follows:

- Data Structure: A fixed length bit vector (i.e., Bloom filter) initialized to zero.
- Recording Operation: A node sets the bits corresponding to its ID (or the incoming edge's ID) in the Bloom filter.
- Path Reconstruction: All possible node/edge IDs are tested for membership in the received Bloom filter. To remove false positives, all possible subsets of nodes/edges are tested until the aggregate signature matches (see Figure 2).

A key parameter for the Bloom filter is its size. Larger Bloom filters yield lower false positive rates and thus decrease the number of required aggregate signature tests. The Bloom filter that records edges is suitable for network coding.

V. Evaluation

We evaluate the performance of the five methods described above in terms of the space requirements and reconstruction time. We first discuss the analytic evaluation and then show quantitative results for a particular parameter space.

A. Analysis

In this analysis, we determine the space requirement, S, and reconstruction time, R, for each method. We use the following parameters:

- Number of unique nodes in the network: n.
- Number of hops traversed by packet: h.
- For Method IV (Sampling), the number of samples taken by each packet: k.
- For Method V (Bloom filter), the size of the Bloom filter data structure: m.

While we obtain the space requirements exactly, we only determine the order of complexity for the reconstruction.

For Method I (Node-Append), we can easily see that the number of bit required for each ID is $\lceil \log_2 n \rceil$. To reconstruct the path, we simply need to traverse the h IDs in the data structure. Thus, we get

$$S_I = \lceil \log_2 n \rceil \times h \quad \text{and} \quad R_I = \mathcal{O}(h)\,.$$

For Method II (Bit Vector), we distinguish between recording nodes and edges. For nodes, a bit vector with one bit per unique node is necessary. To reconstruct the list of nodes, this vector needs to be traversed. Thus, we obtain

$$S_{II,node} = n \quad \text{and} \quad R_{II,node} = \mathcal{O}(n)\,.$$

For edges, we use a bit vector with a bit for each of the $n \times (n-1)$ edges in the graph. Thus,

$$S_{II,edge} = n \times (n-1) \quad \text{and} \quad R_{II,edge} = \mathcal{O}\big(n^2\big)\,.$$

For Method III (Prime Number IDs), the space requirements depend on the size of the prime numbers required in the network. Since this depends on the number of nodes, we introduce a function $\psi(n)$, which provides the n^{th} smallest prime number: $\psi(n) = \{min(x) \mid \pi(x) \geq n\}$, where $\pi(x)$ is the prime counting function. To store the product of h prime numbers, we need $\lceil \log_2[\psi(n) \times \psi(n-1) \times \cdots \times \psi(n-h+1)] \rceil$ bits. Thus, we obtain

$$S_{III,node} = \lceil \log_2[\psi(n) \times \psi(n-1) \times \cdots \times \psi(n-h+1)] \rceil.$$

For edges, the number of unique IDs is $\mathcal{O}\big(n^2\big)$ since each node can be connected to each other node. Thus, the space requirement is

$$S_{III,node} = \lceil \log_2[\psi(n^2) \times \psi(n^2-1) \times \cdots \times \psi(n^2-h+1)] \rceil.$$

The reconstruction time is the time that it takes to factorize a product with $S_{III,node}$ or $S_{III,edge}$ bits respectively. We need to try possibly all n possible IDs for nodes (and n^2 possible IDs for edges). Thus, the reconstruction time is

$$R_{III,node} = \mathcal{O}(n) \quad \text{and} \quad R_{III,edge} = \mathcal{O}\big(n^2\big)\,.$$

For Method IV (Sampling), the space requirement depends on how many IDs are sampled per packet (parameter k):

$$S_{IV,node} = k \times \lceil \log_2 n \rceil \quad \text{and} \quad S_{IV,edge} = k \times \lceil \log_2 n^2 \rceil$$

The reconstruction time is the expected number of packets needed to retrieve h unique samples. With each node/edge adding its ID with probability $p = 1/h$, we obtain:

$$R_{IV,node} = R_{IV,edge} = \mathcal{O}\left(\frac{\ln h}{k \times 1/h \times (1-1/h)^{h-1}}\right).$$

Note that the reconstruction time, R_{IV}, is measured in number of packets required rather than computational operations.

For Method V (Bloom Filter), we use parameter m to determine the size of the data structure. Thus,

$$S_{V,node} = S_{V,edge} = m.$$

The reconstruction time depends on two factors: (1) the time it takes to extract nodes/edges from the Bloom filter and (2) the time it takes to remove false positives by checking the aggregate signature. To obtain the nodes/edges, all nodes/edges are simply tested for membership. This requires $\mathcal{O}(n)$ or $\mathcal{O}\big(n^2\big)$ operations respectively. (Since the complexity of a signature check dominates the extraction computation, we do not further consider the overhead to extract nodes and edges.) The number of false positives in the Bloom filter with l hash functions are (see [12]):

$$fp(m,l,h) = \left(1 - \left(1 - \frac{1}{m}\right)^{lh}\right)^{l}.$$

The binomial distribution can give us the probability of obtaining exactly j false positives when checking all nodes or edges:

$$P[j \text{ false positives}] = \binom{n'}{h} \times fp^j \times (1-fp)^{n'-j},$$

where fp has the appropriate parameters and $n' = n-h$ (since there are at most $n-h$ false positives). For j false positives, there are $h+j$ nodes/edges to choose from and exactly h are correct. Thus, $\binom{h+j}{h}$ choices need to be tested with the aggregate signature. Summing over all possible values of j, we obtain:

$$R_V = \mathcal{O}\left(\sum_{j=0}^{n'} \binom{n'}{j} \times fp^j \times (1-fp)^{n'-j} \times \binom{h+j}{h}\right),$$

where $n' = n-h$ for nodes and $n' = n^2-h$ for edges. Note that the reconstruction time, R_V, is measured in number of signature checks rather than simple computational operations.

B. Lower Bound

In order to evaluate how close the proposed solutions are to the theoretical optimum, we derive the lower bound on the space requirement. While we may not know a practical method that obtains the lower bound, we know that such a function would need to fulfill the following requirement: Every possible combination of node/edge IDs should map to a distinct element in the image set and the size of image set should be minimal (i.e. injective if not bijective).

For n nodes and h hops, there are $\binom{n}{h}$ possible paths of length h. In addition the number of all possible paths consisting of h or less nodes is $\sum_{k=1}^{h} \binom{n}{k}$. A candidate function thus should map the set of these combinations to some image set Y, where $|Y| \geq \sum_{k=1}^{h} \binom{n}{k}$. Thus, the minimum possible size of the lower bound for unordered path recording, $S_{opt,unordered}$ is

$$S_{opt,unordered} = \lceil \log_2(\sum_{k=1}^{h} \binom{n}{k}) \rceil.$$

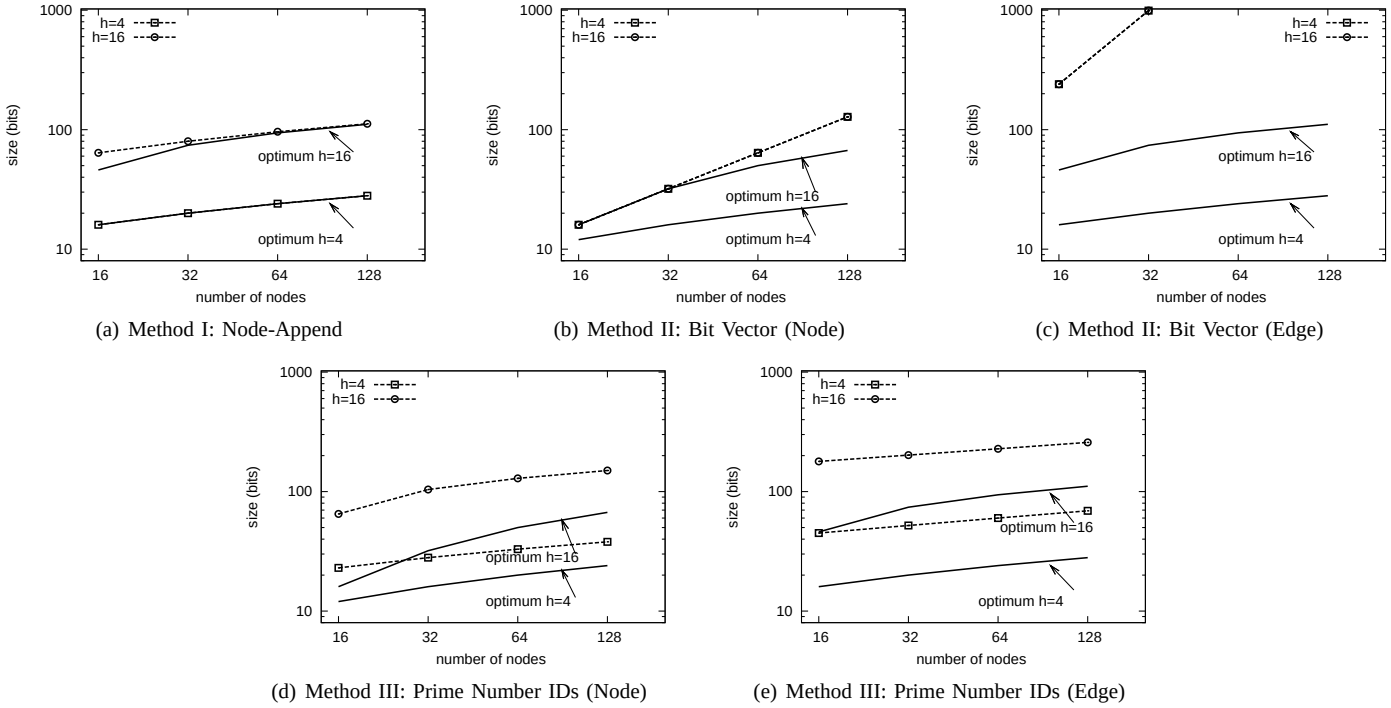
Similarly, the minimum possible size for ordered reconstruction depends on the number of path permutations. For up to h hops, the minimum size, $S_{opt,ordered}$, is

$$S_{opt,ordered} = \lceil \log_2(\sum_{k=1}^{h} P(n,k)) \rceil,$$

where $P(n,k)$ is the number of permutations of length k from n choices.

C. Results

With the above analysis, we explore the quantitative trade-offs between different methods. Figure 4 shows the size requirements for all deterministic methods and compares them to the respective lower bound ("optimum"). Two lines are shown, one for $h = 4$ hops and one for $h = 16$ hops. The following observations can be made:



(a) Method I: Node-Append (b) Method II: Bit Vector (Node) (c) Method II: Bit Vector (Edge)

(d) Method III: Prime Number IDs (Node) (e) Method III: Prime Number IDs (Edge)

Fig. 4. Size Requirements of Deterministic Methods.

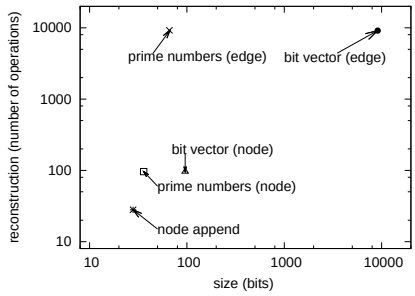
- Method I (Node-Append) approaches the optimum. Its drawback is that is not suitable for network coding, but only for point-to-point paths.
- Method II (Bit Vector) performs well when recording an (unordered) list of nodes. For edges, it does not perform well.
- Method III (Prime Number IDs) is near the optimum in the limit. I can be used both for an unordered list of nodes or for network coding.

For probabilistic path recording methods, the data structure size depends on parameters chosen by the user and thus cannot be shown in graphs.

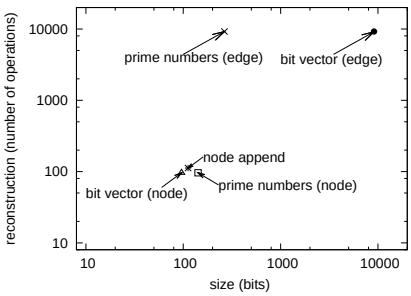
In terms of reconstruction cost, we show the relationship between computational requirements for reconstruction versus the data structure size in Figure 5. The number of nodes is fixed at $n = 96$ and the number of hops is again $h = 4$ and $h = 16$. We make the following observations:

- Method I (Node Append) performs well due to low space requirements and easy reconstruction. It is suitable for point-to-point paths, but not for network coding.
- Method II (Bit Vector) performs well for the unordered vector of nodes. For edges, it requires a large size and computational cost, but can provide a complete network subgraph.
- Method III (Prime Number IDs) performs well for larger numbers of hops since the reconstruction time only depends on the number of unique nodes. It is also suitable to record arbitrary subgraphs in network coding.

When considering probabilistic methods, we obtain the results shown in Figures 6 and 7. For Method IV (Sampling),



(a) $h = 4$



(b) $h = 16$

Fig. 5. Reconstruction Cost vs. Size of Deterministic Methods (for $n = 96$ nodes).

Figure 6 shows the expected number of packets required to obtain a complete path record for different values of k (i.e., the number of samples per packet). As expected, larger k values require fewer packets for reconstruction. For Method V (Bloom Filter), Figure 7 shows the number of signature checks

TABLE I
RECOMMENDATIONS FOR PATH RECORD METHODS.

Path record requirements			Best method	Size	Reconstruction
Deterministic	nodes	unicast	unordered: bit vector / ordered: node-append	96 / 112 bits	96 / 112 ops
	edges	net coding	prime number IDs	142 bits	96 ops
	nodes	unicast	node-append	112 bits	112 ops
	edges	net coding	prime number IDs	265 bits	9216 ops
Sampling	nodes	unicast	samples per packet $k = 4$	28 bits	29.2 packets
	edges	net coding	samples per packet $k = 4$	56 bits	29.2 packets
Bloom filter	nodes	unicast	Bloom filter size $m = 512$	512 bits	1.26 sign. checks
	edges	net coding	Bloom filter size $m = 512$	1024 bits	4.11 sign. checks

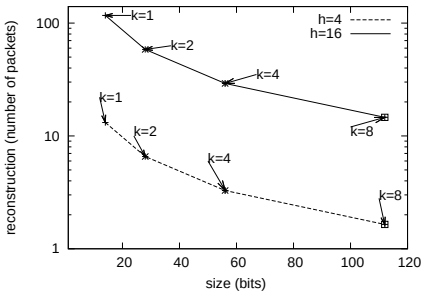


Fig. 6. Reconstruction Cost (Number of Packets) vs. Size for Sampling Method (for $n = 96$ nodes).

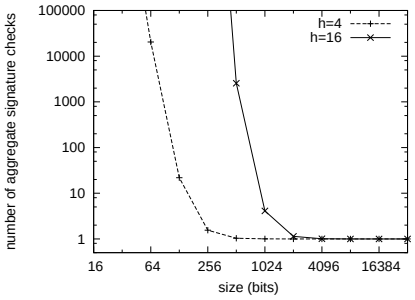


Fig. 7. Reconstruction Cost (Number of Signature Checks) vs. Size for Bloom Filter Method (for $n = 96$ nodes).

for different Bloom filter configurations are considered. Due to the factorial increase in signature checks for increasing false positives, the reconstruction cost is very large for small Bloom filters. With a larger Bloom filter, false positives decrease and only a single check is necessary.

D. Recommendations

To summarize the results of the evaluation of different path record methods, we provide some general recommendations on which method to choose in different cases. We assume a network with $n = 96$ nodes and a maximum of $h = 16$ hops (i.e., network coding). Such a configuration is representative of the "Lakehurst" scenario that is commonly used in MANET evaluations [13].

Table I shows the suggested choices for different path recording requirements. For probabilistic methods (Sampling and Bloom Filter), the recommendation is not a strict optimum since there are tradeoffs as shown in Figures 6 and 7.

VI. SUMMARY

The record of the path of a packet through a MANET can be used to validate control plane routing information. Such a mechanism can defend against malicious attempts to disseminating incorrect connectivity information. We explore different methods for recording the identifiers of nodes and links in the network. We consider both deterministic and probabilistic methods and evaluate their performance in terms of space requirements and reconstruction cost. Our evaluation shows the quantitative tradeoffs between these methods.

REFERENCES

[1] *Network Centric Warfare*, Department of Defense, Washington, DC, Jul. 2001, report to Congress.

[2] *Global Information Grid Architectural Vision*, Department of Defense, Washington, DC, Jun. 2007.

[3] H. Ballani, Y. Chawathe, S. Ratnasamy, T. Roscoe, and S. Shenker, "Off by default!" in *Proc. of Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, College Park, MD, Nov. 2005.

[4] T. Wolf, "A credential-based data path architecture for assurable global networking," in *Proc. of the 2007 IEEE Conference on Military Communications (MILCOM)*, Orlando, FL, Oct. 2007.

[5] N. G. Duffield and F. Lo Presti, "Network tomography from measured end-to-end delay covariance," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 978–992, Dec. 2004.

[6] H. Tian and H. Shen, "Multicast-based inference of network-internal loss performance," in *Proc. of 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 2004)*, Hong Kong, China, May 2004, pp. 288–293.

[7] ——, "Mobile agents based topology discovery algorithms and modelling," in *Proc. of 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 2004)*, Hong Kong, China, May 2004, pp. 502–507.

[8] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[9] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. of International Conference on the Theory and Applications of Cryptographic (EUROCRYPT 2003) (Lecture Notes in Computer Science)*, vol. 2656. Warsaw, Poland: Springer Verlag, May 2003, pp. 416–432.

[10] A. S. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-based IP traceback," in *Proc. of ACM SIGCOMM 2001*, San Diego, CA, Aug. 2001, pp. 3–14.

[11] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[12] A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," in *Proc. of the 40th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, Oct. 2002, pp. 636–646.

[13] D. Caprioni and A. Russo, "Small unit operations situation awareness system (SUO-SAS) radio architecture and system field testing results," in *Proc. of the 2003 IEEE Conference on Military Communications (MILCOM)*, Monterey, CA, Oct. 2003, pp. 198–203.