JOHN WILEY & SONS, LTD., THE ATRIUM, SOUTHERN GATE, CHICHESTER P019 8SQ, UK

*** PROOF OF YOUR ARTICLE ATTACHED, PLEASE READ CAREFULLY ***

After receipt of your corrections your article will be published initially within the online version of the journal.

PLEASE NOTE THAT THE PROMPT RETURN OF YOUR PROOF CORRECTIONS WILL ENSURE THAT THERE ARE NO UNNECESSARY DELAYS IN THE PUBLICATION OF YOUR ARTICLE

□ **READ PROOFS CAREFULLY**

ONCE PUBLISHED ONLINE OR IN PRINT IT IS NOT POSSIBLE TO MAKE ANY FURTHER CORRECTIONS TO YOUR ARTICLE

§ This will be your only chance to correct your proof

§ Please note that the volume and page numbers shown on the proofs are for position only

□ **ANSWER ALL QUERIES ON PROOFS** (Queries are attached as the last page of your proof.)

§ List all corrections and send back via e-mail to the production contact as detailed in the covering e-mail, or mark all corrections directly on the proofs and send the scanned copy via e-mail. Please do not send corrections by fax or post

□ **CHECK FIGURES AND TABLES CAREFULLY**

§ Check sizes, numbering, and orientation of figures

§ All images in the PDF are downsampled (reduced to lower resolution and file size) to facilitate Internet delivery. These images will appear at higher resolution and sharpness in the printed article

§ Review figure legends to ensure that they are complete

§ Check all tables. Review layout, titles, and footnotes

□ **COMPLETE COPYRIGHT TRANSFER AGREEMENT (CTA) if you have not already signed one**

§ Please send a scanned signed copy with your proofs by e-mail. **Your article cannot be published unless we have received the signed CTA**

Additional reprint and journal issue purchases

§ Should you wish to purchase additional copies of your article, please click on the link and follow the instructions provided: http://offprint.cosprinters.com/cos/bw/

§ Corresponding authors are invited to inform their co-authors of the reprint options available.

§ Please note that regardless of the form in which they are acquired, reprints should not be resold, nor further disseminated in electronic form, nor deployed in part or in whole in any marketing, promotional or educational contexts without authorization from Wiley. Permissions requests should be directed to mailto: permissionsuk@wiley.com

§ For information about 'Pay-Per-View and Article Select' click on the following link: http://www3.interscience.wiley.com/aboutus/ppv-articleselect.htm

SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2010; **3**:1–27
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/sec.220

RESEARCH ARTICLE

Source–destination obfuscation in wireless *ad hoc* networks

Thaier Hayajneh[1], Razvi Doomun[2], Prashant Krishnamurthy[3] and David Tipper[3]*

[1] The Hashemite University, Zarqa, Jordan
[2] University of Mauritius, Reduit, Mauritius
[3] University of Pittsburgh, Pittsburgh, PA, U.S.A.

ABSTRACT

The identity and/or location of communicating entities in wireless *ad hoc* networks is extremely important due to the potential of their being identified and subsequently subjected to cyber or physical attacks. In this paper, we show that a global attacker who can eavesdrop on the overall data transmissions and count them can simply visualize the transmissions and infer contextual information. Current approaches to obfuscate the locations of source and destinations do not provide protection against such attacks. We propose two novel techniques (1) SECLOUD: Source and Destination Seclusion using Clouds to obfuscate the true source/destination nodes and make them indistinguishable among a group of neighbor nodes, and (2) ANONYRING: Anonymous Ring which hides the source/destination nodes within a group of nodes that form a ring. Both proposed techniques work well even under network-wide traffic visualization by a global attacker. Furthermore the proposed techniques are shown *via* simulation to be superior to existing schemes in the literature. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS
ad hoc; obfuscation; anonymity; privacy; hiding; entropy; visual analytics

***Correspondence**
David Tipper, University of Pittsburgh, Pittsburgh, PA, U.S.A.[Q2]
E-mail: dtipper@mail.sis.pitt.edu

1. INTRODUCTION

Malicious traffic analysis and privacy attacks against communicating nodes are passive and difficult to detect in large wireless *ad hoc* networks [1]. The disclosure of contextual information about network traffic patterns can be dangerous in sensitive application scenarios. For example, in tactical military wireless networks, command centers could be communicating with each other through an *ad hoc* network of intermediate nodes. Analysis of *traffic* in such an environment may reveal the locations of command centers which will enable the adversaries to launch targeted cyber or physical attacks on them. Hence, it is critical to hide the location of the source and destination for quasi-stationary communicating nodes in *ad hoc* networks.

Traffic analysis attacks make use of rate monitoring that involves counting the number of transmitted/received packets around network nodes and time-correlation analysis that involves finding communication patterns by analyzing latencies between packet transmissions around nodes. Both techniques can be used independently or together to determine the communicating source–destination node pairs. Attackers can be broadly classified as either local or global. A local attacker can generally eavesdrop on transmitted packets around one node at a time and does not know the overall network topology. On the other hand, a global attacker [2] is able to visualize the overall network topology and is capable of network-wide traffic rate monitoring and time-correlation analysis.

Recently different methods have been proposed in the literature for defending against traffic analysis and location privacy related attacks in wireless networks [2–5]. These approaches provide some level of protection against attackers by thwarting traffic analysis attacks and misleading attackers with randomized or fake traffic [1–4,6–8]. However, several of these approaches primarily deal with location privacy in wireless sensor networks where the entity to be protected is a sink node [3,5] or specific sensing sources [6–8]. Furthermore, existing solutions to obfuscate the locations of source or destination nodes do not work well when a global attacker exists.

In this paper, we address the problem of hiding the source/destination nodes from an attacker with a complete view of the network topology and traffic. We propose two simple and efficient techniques for protection against such a global adversary: (i) SECLOUD: source and destination Seclusion using Clouds, that hides the source and destination nodes in a group of nodes (called a 'cloud') that are indistinguishable and (ii) ANONYRING: Anonymous source/destination transmission using a Ring communication path that obfuscates the source and destination nodes in a set of identical nodes comprising a ring. The performance of SECLOUD and ANONYRING are evaluated assuming a sophisticated global attacker capable of using network-wide traffic visualization. We compare SECLOUD and ANONYRING with two commonly used obfuscation techniques, Random Walk (RW) and Fractal Propagation (FP) [3,7] using overhead, anonymity, and unlinkability as metrics. Our simulation results and analyses show that SECLOUD and ANONYRING provide better anonymity than either RW or FP schemes in the presence of a global traffic visualization attack, while having a comparable overhead. Additionally, we show that entropy, which is a commonly used metric, is not always an adequate measure of the degree of anonymity provided by a technique.

The rest of the paper is organized as follows. Section 2 describes related work on obfuscation/privacy techniques. The attacker model, network assumptions, and performance metrics are presented in Section 3. The details of the SECLOUD and ANONYRING protocols are explained in Sections 4 and 5, respectively. Section 6 presents the results with comparative performance evaluation and discussions. Section 7 concludes the paper.

2. RELATED WORK

Here, we briefly summarize related literature on hiding source and destination nodes in wireless networks followed by a more detailed description of the Random Walk and Fractal Propagation schemes since we use them for a comparison with our proposed schemes. While these techniques are effective against local attacks, they have not been extensively evaluated under global attack conditions. As we show later, eavesdropping by a global attacker (which can count packets and visualize the network-wide traffic) can expose the source and destination regions. We note here that most of the current research work assumes that all packets are encrypted link-by-link, padded to prevent potential packet type identification through size, and use anonymous routing schemes to avoid detection of routes during route set-up. Energy constraints and overhead are typically not factored because the underlying assumption is that obfuscation requires transmission of dummy packets.

Phantom routing is a source-node location hiding technique used in References [6,9] to protect against the threat of a local attacker. In this technique each packet potentially traverses a different random path of distance H-hops to a phantom source (i.e., a bogus source node) in the network. Then, the phantom source uses either shortest path [6] or localized flooding [9] to send packets to the final destination. This scheme prevents attackers from discovering the source node's true location by eavesdropping on wireless links hop-by-hop. A similar idea is used in Reference [10], where a path confusion algorithm is used to increase source location anonymity under a local adversary model. Mehta *et al*. in Reference [2] proposed a scheme under a global attack model by hiding the real source among $k-1$ fake sources that emulate the information sensing pattern of real sources in a sensor network. In Reference [8], under a global attack model, the authors proposed statistically strong source anonymity by employing network-wide dummy messages to achieve global privacy. In Reference [4], the authors introduce a similar approach with carefully chosen dummy traffic to hide the real event sources in combination with mechanisms to drop dummy messages to prevent an explosion of network traffic. Some sensor nodes act as proxies that proactively filter dummy messages on their way to the base station destination. The amount of dummy traffic and location & number of fake sources are important factors that determine the effectiveness of the aforementioned obfuscation mechanisms.

2.1. Random walk

In Random Walk-based (RW) obfuscation schemes [6,9], packets at a node are forwarded probabilistically to one of the neighbor nodes within transmission range, and this process continues at each node until the packets reach the destination. This method diversifies routing paths in an unpredictable manner, to avoid rate monitoring attacks against a local eavesdropper. The ability to hide sources and destinations is controlled by the probability P_r to forward in the direction of the destination and probability $(1-P_\mathrm{r})$ to randomly forward to one of neighboring nodes [3].

The RW scheme is effective against local attackers with no prior knowledge of the overall network topology and it increases the complexity and attack time to search for the destination in large *ad hoc* networks. However, it generally leads to a longer routing path on average than the shortest path. This in turn incurs a much higher energy consumption and additional delay that is proportional to the extra distance required to forward packets to the destination. The effectiveness of RW depends on the network's node degree, connectivity, and the source–destination distance. Directed Random Walk (DRW) proposed in Reference [9] is an evolved version of RW that makes use of directional information to obfuscate forwarding of packets. DRW considers two groups of neighbor nodes in opposing directions in which one node is randomly picked so that the walk (i.e., packet forwarding path) leaves the source area to reach the destination. However, grouping the neighbor nodes based on direction requires location information. Xi *et al*. [7] proposed another RW improvement, referred as Greedy Random Walk (GROW) scheme, for preserving location privacy of the source node. GROW uses a bi-directional random walk on a per-packet basis from the main source and destination to confuse a local eavesdropper from tracing the communication path. GROW enhances the basic RW by using local broadcasting and filtering. From the source, each node forwards packets randomly by broadcasting to its least used neighbor node for a certain minimum number of hops until it intersects a randomly pre-established receptor node along the destination path at some point. Overall, the RW based schemes show that using erratic per-packet routes is very effective in resisting rate monitoring and traffic analysis attacks by adversaries monitoring a limited set of nodes.

2.2. Fractal propagation

Deng *et al*., [3] proposed fractal propagation (FP) to counteract local rate monitoring and correlation attacks in sensor networks. Fractal propagation overcomes one of the drawbacks of the random walk scheme by introducing fake packets which are spread out to combat time correlation attacks [3]. When a node forwards a real packet, its neighbor nodes can generate a fake packet with probability (P_f). The fake packet is forwarded to a randomly chosen neighbor node each time over a propagation distance of k-hops. The propagation distance k of the fake packets causes network traffic to appear spread out along different routes resulting in tree-like transmission paths that are more diffuse than the random walk method. Thus, a local attacker takes more time to find the path taken by real packets as it cannot differentiate between encrypted real packets and encrypted fake packets. The performance of fractal propagation can be adjusted by varying the parameters P_f and k. Higher traffic randomness and path confusion is achieved by increasing P_f and k, but the overhead/energy cost will also increase rapidly as well, as more packets collide and higher packet loss rates occur, especially near the destination node.

3. MODELS AND ASSUMPTIONS

3.1. Network model

Our assumptions are consistent with most of the related work in the literature. Specifically, we consider an *ad hoc* network with nodes being distributed either in a grid like manner or randomly. All the MAC and routing protocol messages are assumed to be encrypted so that no leakage of information occurs to the adversary. The nodes' MAC address, IP address and node IDs are also hidden and not advertised. In Reference [11] the authors used short-lived disposable MAC addresses to prevent the real node IDs from being revealed to adversaries. A similar technique is assumed here to avoid identification of nodes.

We assume the existence of a key management protocol that can distribute pair-wise keys between nodes or public-secret key pairs for each node [12,13]. Any of these schemes can be used to set up pairwise keys and authenticate node relationships and we omit the details here. Each packet is encrypted and authenticated so that an adversary cannot decrypt or modify the contents of an eavesdropped packet transmission. All packets are transmitted in the same format and have same length (by padding or fragmenting). Finally, route discovery communications are assumed to be anonymous using any of the anonymous routing protocols such as in References [14–18]. An anonymous routing protocol allows neighbor nodes to authenticate each other without revealing their identities. For example, in Reference [14] the anonymous neighbor authentication is based on dynamically changing pseudonyms of nodes instead of their real identifiers or MAC addresses. Anonymous route discovery and data forwarding employs pairwise *shared link identifiers* between neighbor nodes which are created and established during neighborhood authentication.

3.2. Attack model

An external, global attacker model is assumed in this paper that has the combined capabilities of different existing attack models as described in References [2,3,19]. The attacker has complete knowledge of the network topology and can keep statistical measurements for all of the network traffic. We assume that the global attacker can perform rate monitoring and time correlation analysis for all traffic in the network (which is a stronger attack than corresponding ones assumed elsewhere). The attacker can visualize statistics (e.g., packet count in a time window) of all transmitted/received packets in the network and determine the traffic density on every link in the network. Furthermore, we assume the attacker can use advanced visualization techniques such as edge detection image processing algorithms to possibly extract the traffic pattern. Details are provided in Section 6.

A possible method for the global attack above is deploying an overlay/underlay network with several malicious nodes simply to passively sense traffic from the given *ad hoc* network, similar to the idea in Reference [2]. These nodes can collect information and collaborate with a centralized entity using a different band.

3.3. Evaluation metrics

We analyze the performance of SECLOUD, ANONYRING, Random Walk, and Fractal Propagation using the metrics—anonymity, unlinkability [19], entropy, and the overhead incurred by the technique.

3.3.1. Anonymity.

Anonymity means hiding information about which node behaves as the source or the destination. The level of anonymity λ is defined as the probability that a node of interest is incorrectly identified in an anonymous group [19]. If a node is hidden among a set A of nodes that have identical behavior, then the level of anonymity is $\lambda = 1 - (1/|A|)$ where $|A|$ is the size of a set A. Thus, the anonymity level of a node (source or destination in our case) depends on the number of nodes in the anonymous zone. If the source node is hidden in a set of nodes A_S and the destination is hidden in a disjoint set of nodes A_D then the anonymity of the source–destination pair is given by $\lambda = 1 - (1/|A_S|)(1/|A_D|)$. Untraceability is a similar notion that indicates how difficult it is for an adversary to identify the packet transmissions from a source based on receptions at the destination.

3.3.2. Unlinkability.

Unlinkability is a generalization of the notion of anonymity and untraceability, (i.e., hiding any contextual information about the relationship between source–destination pairs). We employ a 2-D packet count matrix and a 3-D graph of transmitted data around nodes to determine whether or not a global attacker can visualize the existence of communication between a source and destination and thus link them.

3.3.3. Entropy.

Entropy is a common metric for quantifying uncertainty in information theory and is often used as a measure of 'hiding' or 'privacy' [3] since an increase in entropy is indicative of an increase in uncertainty in information. An entropy-based measure of privacy is given as $H = -\sum_i p_x \log_2 p_x$ where p_x is the probability for the attacker that the observed data transmissions were from node x in the anonymity set A of size $|A|$. It can be normalized to obtain a privacy metric value between 0 (no privacy) and 1 (guaranteed perfect privacy), as $\epsilon = H/H_{\max}$, where $H_{\max} = -\log_2 p_x$ with $p_x = |A|$. While entropy is an appealing concept providing a scalar number for privacy, it does not capture the amount of contextual information available that can disclose the location of source and destination nodes. For example, most nodes may be sending roughly similar numbers of packets. This results in a high average entropy. But the few nodes that send more packets may be sufficient to reveal the identity or location of a source or destination node. We will show in our simulation based results that an increasing entropy value does not necessarily imply that the source–destination nodes are adequately hidden.

3.3.4. Overhead.

We define the transmission overhead factor (TOF) as the ratio of total number of packets transmitted by nodes using an obfuscation technique to the total number of transmissions without the obfuscation technique using simply shortest path routing between source and destination nodes. The transmission overhead is also directly related to the energy consumption of nodes when using the obfuscation mechanisms for communication.

4. THE SECLOUD PROTOCOL

The general idea of SECLOUD, is to seclude the source and destination node locations within a cloud of irregular shape that is constructed using its neighboring nodes. The details of this protocol are explained in the following.

Let S denote the source node and D denote the destination node of the communicating node pair. The protocol first conducts neighborhood discovery to form the potential cloudy region around the source and destination. The source node S first broadcasts a hello message to discover all its one-hop neighbors $N_S(1, i)$ for $i = (1, 2, \ldots m)$, where m is the total number of neighbor nodes. Then, the nodes in $N_S(1, i)$ discover their respective neighbors $N_S(2, i)$ which are two-hops away from node S. Consequently, source node S constructs the list: $R_S = \{N_S(1, i), N_S(2, i), N_S(3, i), \ldots N_S(k, i)\}$, where $N_S(k, i)$ is the set of kth hop neighbors of node S. This initialization process of neighbor discovery is done periodically by all nodes i in the network, for $(1 \le i \le n)$. This will ensure that the attacker cannot determine which of the nodes performing the initialization will be the source.

Once the source node has identified the set R_S of k nearest neighbors, it forms a cloud by first selecting the center point of the cloud N_o arbitrarily from R_S. As an example, let $k = 3$. The source node S will then request node N_o to send its set of k neighbor nodes $R_o = \{N_o(1, i), N_o(2, i), N_o(3, i)\}$. It will randomly select a set of nodes, B for the cloud, such that $B \subseteq R_o$. This will keep the cloud region irregular—the source cannot be predicted to be in the center of the cloud for example. The nodes in B will be marked as pseudo-sources in the cloud and are requested to transmit encrypted dummy packets at a rate similar to the source transmission rate. Dummy packets are simply dropped. The destination node D will also follow the same initialization procedure to construct a cloud. *Note the size of the source and the destination clouds can be different by using different values of k and B for each, depending on the obfuscation strength needed on each side.*

Let B_S denote the size of the source cloud drawn from a set of k_S hop neighbors and B_D denote the size of the destination cloud which is drawn from a set of k_D hop neighbors. To further hide the source and destination in their respective clouds we use the concept of a delegated source and destination which is similar to the phantom source concept of Kamat [6]. Node S randomly selects one or more nodes from the set B_S to act as *delegated sources*. Similarly D randomly selects one or more nodes in its cloud B_D to act as *delegated destinations*. The delegated sources are required to have connectivity (typically multi-hop) with the source S. Similarly the delegated destinations must have connectivity with the destination D. Real packets are relayed locally from S to the delegated sources. A delegated source node will set up route(s) to forward real packets to reach a delegated destination node. Thus, real packet traffic will 'move' from the source cloud to the destination cloud *via* the route(s) between a delegated source–destination pair. After reaching the delegated destinations, real packets are forwarded locally to the actual destination D. Delegated sources find routes to delegated destinations using a suitable anonymous routing protocol (e.g., see References [14,15]). The delegated source–destination nodes hide the communications of the real source with the real destination. In the event a delegated source or destination node is attacked, the protocol can include options to chose and set-up another delegated source–destination pair to resume communications through a new route. If we do not use delegated source–destination nodes, and if the source from one cloud sends packets directly to the destination in the other cloud, then the route between the source and destination may possibly be disclosed.

With the above set-up, SECLOUD achieves source/destination privacy in a local region. In a single source–destination scenario, the global attacker will have to guess the source from the set of nodes B_S and the destination from the set of nodes B_D. Assuming that B_S and B_D are disjoint, SECLOUD results in the communicating pair anonymity level of $\lambda = 1 - (1/B_S)(1/B_D)$. To further improve the hiding of sources/destinations one can simply increase the size of the clouds (B_S, B_D) used. Additionally, SECLOUD can create fake sources and fake destination clouds as well. The fake sources/destination clouds will behave in a manner similar to the real source destination clouds and will communicate with each other. Fake clouds increase the likelihood of misleading the attacker who may attack (e.g., by jamming) the fake clouds instead of the real clouds. Obviously the fake clouds will improve the anonymity but at the expense of additional overhead. Specifically, with f disjoint fake clouds each using the same cloud sizes as the true source and destination B_S and B_D, the anonymity becomes $\lambda = 1 - (1/(f+1))(1/(B_S))(1/(B_D))$.

The transmission overhead factor of SECLOUD for the single source–destination scenario assuming only a single delegated source–destination pair can be estimated as follows. Let L be the length of the shortest path between S and D and L_B be the length of the path between the source S through a delegated source–destination pair to the destination D. Then the transmission overhead factor—TOF—consists of the dummy packets transmitted within the clouds and the additional transmissions arising from using the longer route through a delegated source–destination pair. Assuming all nodes involved in SECLOUD transmit the same number of packets, TOF can be approximated as $\text{TOF} \approx (B_S + B_D + L_B)/L$. If we use f fake clouds and each fake cloud transmits the same number of packets as the real clouds, the transmission overhead factor becomes $\text{TOF} \approx (f+1)(B_S + B_D + L_B)/L$.

5. THE ANONYRING PROTOCOL

The general idea of ANONYRING (anonymous ring) is to obfuscate the source and destination node locations with a ring of nodes constructed through the source and destination.

ANONYRING begins with a discovery procedure like the one used in SECLOUD. All nodes broadcast a hello message to discover all their one-hop neighbors $N(1, i)$ for $i = 1, 2, \ldots, m$, where m is the total number of one hop neighbor nodes. Then, the nodes in $N(1, i)$ discover their respective neighbors $N(2, i)$. Consequently, each node will construct the list: $\{N(1, i), N(2, i), N(3, i), \ldots N(k, i)\}$, where $N(k, i)$ is the set of its kth hop neighbors. This initialization process of neighbor discovery is done periodically by all nodes in the network.

Once the discovery phase is over the source node initiates ring formation. Several algorithms exist in the literature for connecting a set of nodes into a ring topology [20,21]. We illustrate this with a simple successive disjoint paths procedure. Specifically, the source node S finds the shortest direct path to the destination node using an anonymous routing protocol. Let the shortest route be L hops long and consist of the nodes $\{N_{s_1}, N_{s_2}, \ldots N_{s_i}, \ldots N_{s_{L-1}}\}$. The source node S then repeats the routing procedure to the destination but not considering the nodes already found in the shortest path route. That is, S finds a second route which is M hops long $\{N^2_{s_1}, N^2_{s_2}, \ldots N^2_{s_i}, \ldots N^2_{s_{M-1}}\}$ that is node disjoint with the first route. Obviously the two routes when put together will form a ring containing S and D and consisting of $\text{RS} = L + M$ nodes. Note than one can choose to increase the ring size RS by selecting longer disjoint routes between the source and destination (i.e., instead of the two shortest disjoint routes could pick the two longest disjoint routes or some combination in between) at the expense of additional overhead. In our simulations, we try to pick rings by moving a few hops beyond the shortest routes in order to increase the anonymity.

In ANONYRING all nodes in RS (excluding nodes S and D) will be marked as pseudo-sources on the ring and are requested to transmit encrypted dummy packets at a rate similar to the source transmission rate and to forward real packets when available from node S to node D. Dummy packets are dropped by receiving neighbor nodes. Note the encrypted real packets can be broadcasted either clockwise or counterclockwise or both ways from source S to reach destination D depending on the required level of security and transmission reliability.

With the above set-up, ANONYRING achieves source/destination privacy within a ring of RS nodes. In single source–destination case, a global attacker will have to guess the source or destination from any of the RS ring nodes. For multiple source–destination scenarios, several rings can be constructed for each source–destination pair. To improve the efficiency of ANONYRING, if possible, a larger common ring can be used to blend different source–destination pairs. For a single source–destination pair since the set of R nodes hiding the source and destination are not disjoint the anonymity is given by $\lambda = 1 - 1/\binom{RS}{2}$. This assumes that the global attacker does not know where the source and destination nodes are located the ring. In fact, they could be adjacent nodes. The transmission overhead factor (TOF) of ANONYRING is approximately $\text{TOF} \approx \text{RS}/L$, where L is the length of the shortest path between S and D.

6. COMPARATIVE PERFORMANCE EVALUATION

In order to evaluate the effectiveness of the RW, FP, SECLOUD, and ANONYRING schemes we used NS-2 to simulate an *ad hoc* network of 400 nodes distributed in an area of 2000 m × 2000 m. Two different node distribution models were used in the simulations: perturbed grid distribution and random distribution. In the perturbed grid case, nodes were located in a perturbed 20×20 grid. The coordinates of each node $(x_i,\ y_j)$ were randomly chosen using uniform random variables in the ranges $(100i - p100, 100i + p100)$ and $(100j - p100, 100j + p100)$, where p is the perturbation parameter and $i = 1, \ldots 20$ and $j = 1, 2, \ldots, 20$, respectively. For the random node distribution case, the coordinates of the nodes (x_i, y_i) for $i = 1, 2, \ldots, 400$ were independently and randomly chosen in the range from 0 to 2000 m using a uniform $[0 - 2000]$ random number generator.

After the nodes were distributed, we employed the Quasi-Unit disk graph (Q-UDG) model [22] to determine the connectivity between nodes and the network topology. In the Quasi-UDG model, a link exists between two nodes if the inter-nodal distance d is less than αR, where R is the transmission range of the node and α is the Q-UDG factor $(0 \leq \alpha \leq 1)$. For distances d greater than R, there is no link connectivity. However, for $\alpha R \leq d \leq R$, the link will exist with probability $(R - d)/(R - \alpha R)$. In our simulation we set $\alpha = 0.2$ and $R = 145$m. We consider only a single source–destination pair to make it easier for the attacker to compromise the privacy. The source node is randomly chosen from the region $(100\,\text{m} \leq \text{x} \leq 900\,\text{m})$ and $(100\,\text{m} \leq \text{y} \leq 900\,\text{m})$ and the destination node is randomly selected from $(1100\,\text{m} \leq \text{x} \leq 1900\,\text{m})$ and $(1100\,\text{m} \leq \text{y} \leq 1900\,\text{m})$ of the network. The source sends 5000 real packets in a time window of T seconds. All simulations are repeated 15 times and results are averaged for each tested scenario.

As noted earlier, we compare the RW, FP, SECLOUD, and ANONYRING based on anonymity, unlinkability, and entropy. We determine unlinkability and anonymity as follows. To visualize linkable communications, the attacker can simply plot a 3-D graph of the number of packets u_i transmitted by each node i during time interval T with the approximate location of each node. Additional processing can help the attacker to better quantify linkability. The attacker will sample n of the nodes that have the highest number of packets transmitted during the time interval T and computes the average value U of packets transmitted. Then nodes that transmit at least βU packets are marked where $0 < \beta < 1$. A 2-D or 3-D graph of nodes, the number of packets transmitted and the *marked nodes* are used to determine possible communication paths, sources, and destinations. We pick $n = 10$ in our simulations. The values of n and β will create sharp or fuzzy boundaries in the graph. Based on these boundaries, we count the size of $|A|$—the number of nodes within which the source and destinations are hidden and use it to determine the anonymity. In the case of the RW scheme, we do not include the nodes at the boundary of the network in picking n or marking nodes, as this would cause bias due to packet transmissions bouncing back.

Another approach to determine the linkability is to convert the packet count information into an image and use an edge detection algorithm to reveal source/destination locations as well as the communication route. For the edge detection, a 2-D matrix of packet counts of node transmissions is formed and is treated as a matrix of pixels. The pixel intensities are normalized between 0 and 255 representing the packet count at a node. We apply the Canny edge detection [23] image processing algorithm to extract the contour of traffic flows from the pixel information. The algorithm first smoothes the image to eliminate noise pixel intensities and finds the image gradient to highlight regions with high spatial derivatives. It then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient matrix is further reduced by hysteresis thresholding. The resulting shape of the contour observed from traffic pattern analysis is used to deduce the location of traffic sources and destinations if possible.

Determination of the entropy h is straightforward as one simply counts the number of packets u_i at each node i and the total number of packets V transmitted in the network during time interval T. The fraction of packets sent by i is $p_i = u_i/V$ and the entropy is defined as $H = -\sum_i p_i \log_2 p_i$.

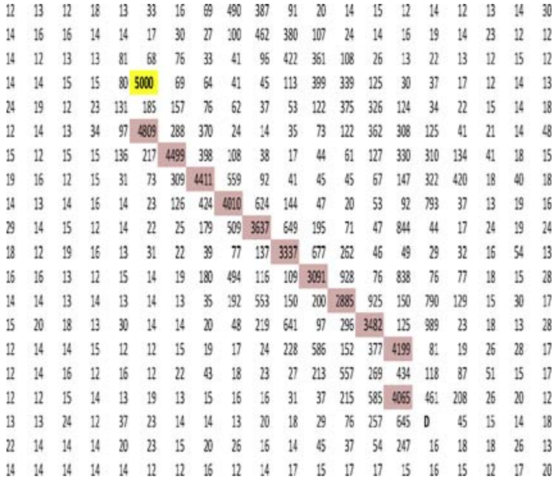
We discuss typical simulation results for each scheme (i.e., RW, FP, SECLOUD, and ANONYRING) in turn below for the perturbed grid network topology. For the sake of brevity we do not include results for the random topology as the conclusions are similar for both topologies.

6.1. Random walk technique

With random walk (RW) [3,6], packets are forwarded in a random fashion from one hop to the next until they reach the destination. A probability P_r is used at each hop to decide how random the forwarding is, where $(0.5 \leq P_\text{r} \leq 1.0)$. If $P_\text{r} = 1.0$, there is no randomness and the packet is sent to next hop node on the shortest path to the destination. We varied the RW parameter $(1 - P_\text{r})$ in steps of 0.05 from $0.05 \leq (1 - P_\text{r}) \leq 0.5$.

In Figure 1, a matrix of the number of packets u_i transmitted by each node i is shown for random walk with $(1 - P_\text{r}) = 0.1$ (sample of one simulation). Nodes are not located exactly as shown but their relative positions are maintained in the matrix. The source node S (yellow color) sends 5000 packets to destination node D (blue color). Using $n = 10$, $\beta = 0.5$, we find $\beta U = 2072$ and mark all nodes transmitting more than βU packets. As shown in Figure 1, the entire route is revealed. The edge node at the source is even more apparent as it will have a global maximum (highest traffic density node). The destination is one-hop away from the next highest traffic node with packet count 4065. Surprisingly the normalized entropy for the

DOI: 10.1002/sec

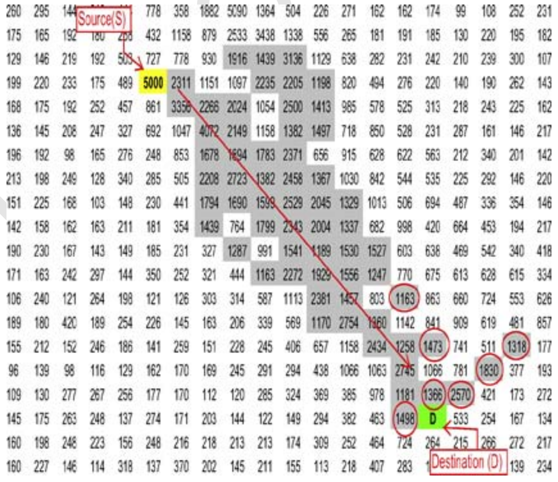


12	13	12	18	13	33	16	69	490	387	91	20	14	15	12	14	12	13	14	30
14	16	16	14	14	17	30	27	100	462	380	107	24	14	16	19	14	23	12	12
14	12	13	13	81	68	76	33	41	95	422	361	108	26	13	22	13	12	15	12
14	14	15	15	80	5000	69	64	41	45	113	399	339	125	30	37	17	12	14	13
24	19	12	23	131	185	157	76	62	37	53	122	375	326	124	34	22	15	14	18
12	14	13	34	97	4809	288	370	24	14	35	73	122	362	308	125	41	21	14	48
15	12	15	15	136	217	4499	398	108	38	17	44	61	127	330	310	134	41	18	15
19	16	12	15	31	73	309	4411	559	92	41	45	45	67	147	322	420	18	40	18
14	13	14	16	14	23	126	424	4010	624	144	47	20	53	92	793	37	13	19	16
29	14	15	12	14	22	25	179	509	3637	649	195	71	47	844	44	17	24	19	24
18	12	19	16	13	31	22	39	77	137	3337	677	262	46	49	29	32	16	54	13
16	16	13	12	15	14	19	180	494	116	109	3091	928	76	838	76	77	18	15	28
14	14	13	14	13	14	13	35	192	553	150	200	2885	925	150	790	129	15	30	17
15	20	18	13	30	14	14	20	48	219	641	97	296	3482	125	989	23	18	13	28
12	14	14	15	12	12	15	19	17	24	228	586	152	377	4199	81	19	26	28	17
12	14	16	12	16	12	22	43	18	23	27	213	557	269	434	118	87	51	15	17
12	12	15	14	13	19	13	15	16	16	31	37	215	585	4065	461	208	26	20	12
13	13	24	12	37	23	14	14	13	20	18	29	76	257	645	D	45	15	14	18
22	14	14	14	20	23	15	20	26	16	14	45	37	54	247	16	18	18	26	13
14	14	14	14	14	12	12	16	12	14	17	15	17	17	15	16	15	12	17	20

Figure 1. Random walk $P_r = 0.9$.

$(1 - P_r) = 0.1$ case is still quite high at $H = 0.8$ not providing any indication of the poor obfuscation in this case. The correlation/linkability between the source and destination node arises due to the nature of the underlying traffic pattern. We find strong source–destination node pair correlation for low values of probability $(1 - P_r)$ of the random walk.

As the RW parameter $(1 - P_r)$ increases, the traffic pattern is more skewed and the obfuscation is better. In Figure 2, a matrix of the number of packets u_i transmitted by each node i is shown for random walk with $(1 - P_r) = 0.3$. The source node S (yellow color) sends 5000 packets to destination node D (green color). Using $n = 10$, $\beta = 0.4$, we find $\beta U = 1160$ and mark all nodes transmitting more than βU packets. An attacker gets a set of possible paths between the source and destination. The destination node can be guessed to be near nodes with highest u_i's, that is at the circled nodes in Figure 2 with packet counts 2570, 1830, 1498, 1366, 1318, 1473, and 1163. The destination anonymity is 6/7 (i.e., $|A| = 7$), whereas the source node packet count is the maximum in its vicinity and can be easily detected—so it's anonymity is zero.


Source(S)
Destination (D)

260	295	144	[illegible]	[illegible]	778	358	1882	5090	1364	504	226	271	162	162	174	99	108	252	231
175	165	192	180	[illegible]	432	1158	879	2533	3438	1338	556	265	181	191	185	130	220	195	182
129	146	219	192	[illegible]	727	778	930	1916	1439	3136	1129	638	282	231	242	210	239	300	107
199	220	233	175	489	5000	2311	1151	1097	2235	2205	1198	820	494	276	220	140	190	262	143
168	175	192	252	457	861	3358	2266	2024	1054	2500	1413	985	578	525	313	218	243	225	162
136	145	208	247	327	692	1047	4072	2149	1158	1382	1497	718	850	528	231	287	161	146	217
196	192	98	165	276	248	853	1678	1694	1783	2371	656	915	628	622	563	212	340	201	142
213	198	249	128	340	285	505	2208	2723	1382	2458	1367	1030	842	544	535	225	292	146	220
151	225	168	103	148	230	441	1794	1690	1599	2529	2045	1329	1013	506	694	487	336	354	146
142	158	162	163	211	181	354	1439	764	1799	2343	2004	1337	682	998	420	664	453	194	217
190	230	167	143	149	185	231	327	1287	991	1541	1189	1530	1527	603	638	469	542	340	418
171	163	242	297	144	350	252	321	444	1163	2272	1929	1556	1247	770	675	613	628	615	334
106	240	121	264	198	121	126	303	314	587	1113	2381	1457	803	1163	863	660	724	553	626
189	180	420	189	254	226	145	163	206	339	569	1170	2754	1360	1142	841	909	619	481	857
155	212	152	246	186	141	259	151	228	245	406	657	1158	2434	1258	1473	741	511	1318	177
96	139	98	116	129	162	170	169	245	291	294	438	1066	1063	2745	1066	781	1830	377	193
109	130	277	267	256	177	170	112	120	285	324	369	385	978	1181	1366	2570	421	173	272
145	175	263	248	137	274	171	203	144	122	149	294	382	463	1498	D	533	254	167	134
160	198	248	223	156	248	216	218	213	213	174	309	252	464	724	264	215	266	272	217
160	227	146	114	318	137	370	202	145	211	155	113	218	407	283	[illegible]	[illegible]	[illegible]	139	234

Figure 2. Random walk $(1 - P_r) = 0.3$, $\beta = 0.4$.

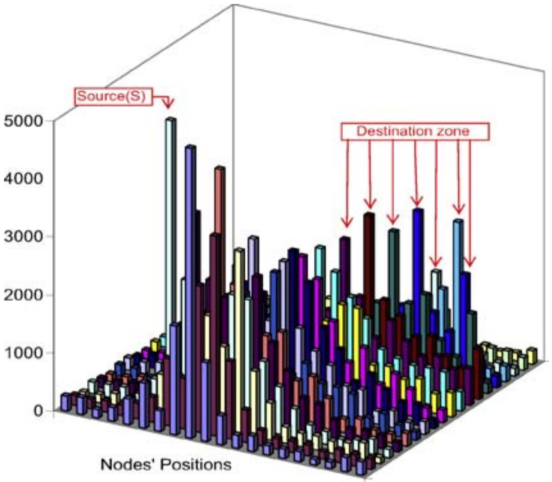


Figure 3. Three dimensional graph of u_i for random walk $(1 - P_r) = 0.3$.

Figure 3 shows a 3-D visualization of the processed global traffic counts u_i and marked nodes of Figure 2, (i.e., $(1 - P_r) = 0.3$, $n = 10$, $\beta = 0.4$). Notice that one can clearly make out the source node and identify a zone of possible locations for the destination.

The output of the edge detection algorithm on the packet count data is shown in Figure 4. Clearly, the attacker can identify the source and the destination with high probability.

The normalized entropy values H for RW are plotted in Figure 5 *versus* $(1 - P_r)$. As one would expect the entropy values increase with decreasing P_r. Notice that the entropy values are high for even low values of $(1 - P_r)$ even though the linkability is quite high as illustrated above. Moreover, the entropy values are very close making it hard to use them meaningfully.

The transmission overhead factor (TOF) is the ratio of total number of packets transmitted with RW over the total

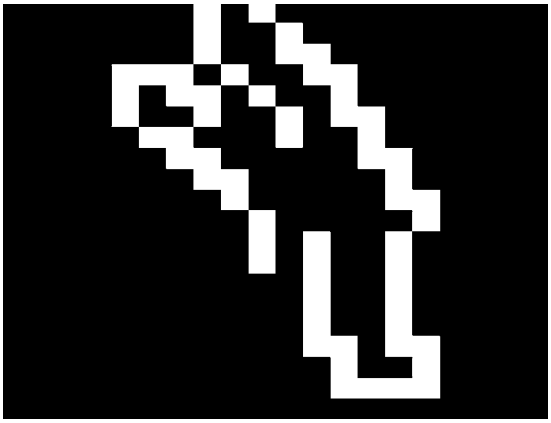


Figure 4. Edge detection output for RW $(1 - P_r) = 0.3$.

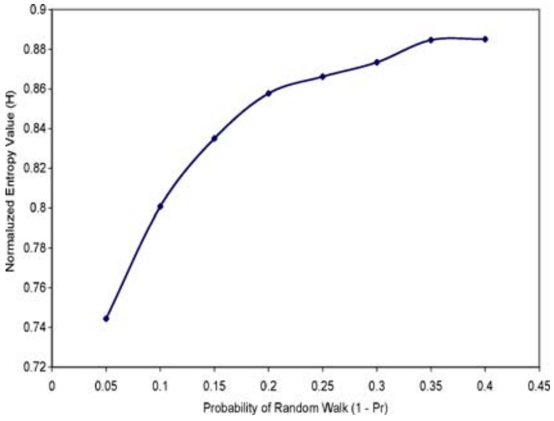


Figure 5. Random walk—normalized entropy.

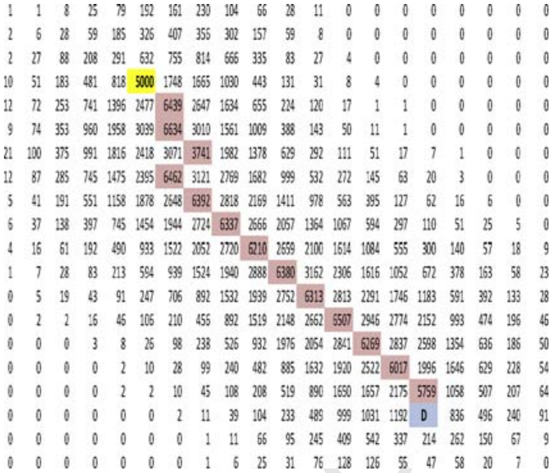
number of packets transmitted with shortest path routing and no RW scheme. Figure 6 shows TOF *versus* $(1 - P_r)$. The overhead is high and the anonymity and unlinkability are not adequate as noted earlier. For example, at $(1 - P_r) = 0.3$ the transmission overhead is approximately 400%.

6.2. Fractal propagation technique

In fractal propagation, neighbor nodes along a forwarding path from source to destination generate fake packets with probability P_{fake}, and the fake packets are propagated K hops by successive neighbor nodes randomly selected. A higher (P_{fake}, K) is expected to enhance communication privacy at the cost of extra overhead.

The results of traffic visualization for one simulation sample by marking nodes ($n = 10$, $\beta = 0.5$, $\beta U = 3197$) is shown in Figure 7 for $P_{\text{fake}} = 0.1$, $K = 5$. We can identify a single source node (yellow color) transmitting 5000 packets to the destination node 13-hops away, denoted by cell D (blue color).

The 3-D traffic graph shown in Figure 8, illustrates a similar result where the peak traffic nodes are aligned. The anonymity is zero since the source–destination locations are both distinguishable by a global attacker. The destination node is just 1-hop away from the node with packet count 5759. Note that the normalized entropy H is quite high at $(H = 0.81)$ though obviously the anonymity and unlinkability to a global attack is poor. As one would expect increasing the rate of fake packets and their propagation distance improves obfuscation. For example, in Figure 9, for the same source and destination nodes, a higher probability of fake packet generation ($P_{\text{fake}} = 0.2$) is used with fractal propagation. The source and destination nodes can both still be located using the same visualization methodology (for $n = 10$, $\beta = 0.5$, in this case $\beta U = 3859$) as before.



1	1	8	25	79	192	161	230	104	66	28	11	0	0	0	0	0	0	0	0
2	6	28	59	185	326	407	355	302	157	59	8	0	0	0	0	0	0	0	0
2	27	88	208	291	632	755	814	666	335	83	27	4	0	0	0	0	0	0	0
10	51	183	481	818	5000	1748	1665	1030	443	131	31	8	4	0	0	0	0	0	0
12	72	253	741	1396	2477	6439	2647	1634	655	224	120	17	1	1	0	0	0	0	0
9	74	353	960	1958	3039	6634	3010	1561	1009	388	143	50	11	1	0	0	0	0	0
21	100	375	991	1816	2418	3071	3741	1982	1378	629	292	111	51	17	7	1	0	0	0
12	87	285	745	1475	2395	6462	3121	2769	1682	999	532	272	145	63	20	3	0	0	0
5	41	191	551	1158	1878	2648	6392	2818	2169	1411	978	563	395	127	62	16	6	0	0
6	37	138	397	745	1454	1944	2724	6337	2666	2057	1364	1067	594	297	110	51	25	5	0
4	16	61	192	490	933	1522	2052	2720	6210	2659	2100	1614	1084	555	300	140	57	18	9
1	7	28	83	213	594	939	1524	1940	2888	6380	3162	2306	1616	1052	672	378	163	58	23
0	5	19	43	91	247	706	892	1532	1939	2752	6313	2813	2291	1746	1183	591	392	133	28
0	2	2	16	46	106	210	455	892	1519	2148	2662	6507	2946	2774	2152	993	474	196	46
0	0	0	3	8	26	98	238	526	932	1976	2054	2841	6269	2837	2598	1354	636	186	50
0	0	0	0	2	10	28	99	240	482	885	1632	1920	2522	6017	1996	1646	629	228	54
0	0	0	0	2	2	10	45	108	208	519	890	1650	1657	2175	5759	1058	507	207	64
0	0	0	0	0	0	2	11	39	104	233	489	999	1031	1192	D	836	496	240	91
0	0	0	0	0	0	0	1	11	66	95	245	409	542	337	214	262	150	67	9
0	0	0	0	0	0	0	1	6	25	31	76	128	126	55	47	58	20	7	0

Figure 7. Fractal propagation $P_{\text{fake}} = 0.1$, $K = 5$.

The output of the edge detection algorithm is shown in Figure 10. The same conclusions as to the location of the source and destination can be extracted by the global attacker from this type of analysis.

In Figure 11, the impact of varying P_{fake} and K on the average normalized network transmission entropy H is shown. The entropy increases as more fake packets (higher

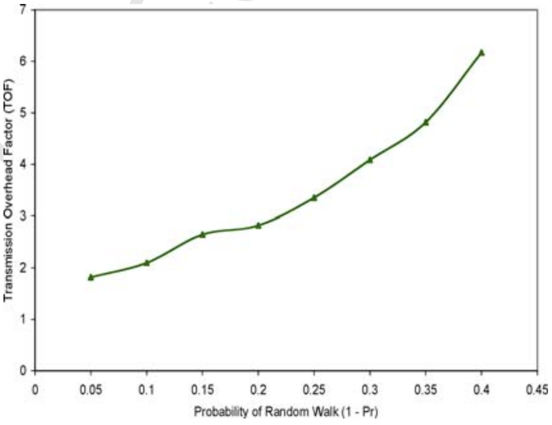


Figure 6. Random walk—transmission overhead factor.

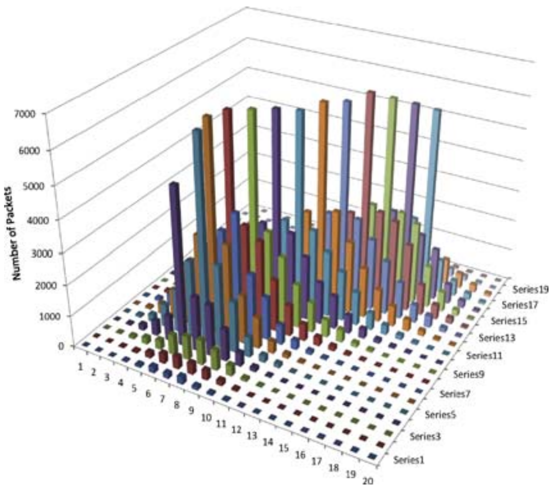
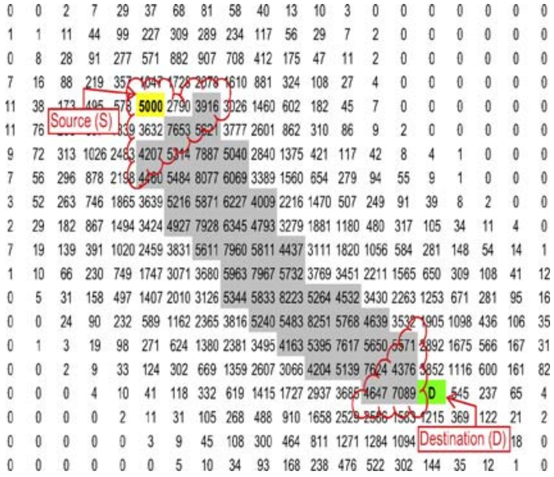


Figure 8. Three dimensional visualization of FP $P_{\text{fake}} = 0.1$, $K = 5$.



0	0	2	7	29	37	68	81	58	40	13	10	3	0	0	0	0	0	0	0
1	1	11	44	99	227	309	289	234	117	56	29	7	2	0	0	0	0	0	0
0	8	28	91	277	571	882	907	708	412	175	47	11	2	0	0	0	0	0	0
7	16	88	219	357	1047	1728	2078	1610	881	324	108	27	4	0	0	0	0	0	0
11	38	173	495	578	**5000**	2790	3916	3026	1460	602	182	45	7	0	0	0	0	0	0
11	76	[illegible]	[illegible]	1839	3632	7653	5621	3777	2601	862	310	86	9	2	0	0	0	0	0
9	72	313	1026	2483	4207	5314	7887	5040	2840	1375	421	117	42	8	4	1	0	0	0
7	56	296	878	2198	4480	5484	8077	6069	3389	1560	654	279	94	55	9	1	0	0	0
3	52	263	746	1865	3639	5216	5871	6227	4009	2216	1470	507	249	91	39	8	2	0	0
2	29	182	867	1494	3424	4927	7928	6345	4793	3279	1881	1180	480	317	105	34	11	4	0
7	19	139	391	1020	2459	3831	5611	7960	5811	4437	3111	1820	1056	584	281	148	54	14	1
1	10	66	230	749	1747	3071	3680	5963	7967	5732	3769	3451	2211	1565	650	309	108	41	12
0	5	31	158	497	1407	2010	3126	5344	5833	8223	5264	4532	3430	2263	1253	671	281	95	16
0	0	24	90	232	589	1162	2365	3816	5240	5483	8251	5768	4639	3532	1905	1098	436	106	35
0	1	3	19	98	271	624	1380	2381	3495	4163	5395	7617	5650	5571	3892	1675	566	167	31
0	0	2	9	33	124	302	669	1359	2607	3066	4204	5139	7624	4376	5852	1116	600	161	82
0	0	0	4	10	41	118	332	619	1415	1727	2937	3685	4647	7089	D	545	237	65	4
0	0	0	0	2	11	31	105	268	488	910	1658	2529	2566	1563	1215	369	122	21	2
0	0	0	0	0	3	9	45	108	300	464	811	1271	1284	1094	[illegible]	[illegible]	[illegible]	18	0
0	0	0	0	0	0	5	10	34	93	168	238	476	522	302	144	35	12	1	0

Figure 9. Fractal propagation $P_{\text{fake}} = 0.2$.

P_{fake}) are propagated. Similarly, for a fixed probability of fake packet generation, the average entropy is higher when the distance K of the fake packet propagation is larger. While the mean entropy values increase with P_{fake} and K, they provide no meaningful information about the ability to obfuscate the source and destination.

The overhead cost with FP is shown in Figure 12. TOF here, as before, is the ratio of the total number of packet transmissions using FP to that with shortest path routing and no FP scheme. The transmission overhead increases significantly for higher P_{fake} and K, as depicted in Figure 12. The extra overhead transmission cost is more compared to the relative increase in entropy values for corresponding P_{fake} and K. Obviously as P_{fake} and K increase, the overhead grows significantly. Even $P_{\text{fake}} = 0.2$ and $K = 5$ results in a high transmission overhead factor (TOF) of 1159%, but the anonymity and unlinkability are poor.

In comparison with fractal propagation, the unlinkability of RW when P_{r} is decreased, is better than FP when it's parameters P_{fake} and K are increased. However in both cases the normalized entropy is misleading in assessing the ease in locating the source and destination nodes that are communicating.

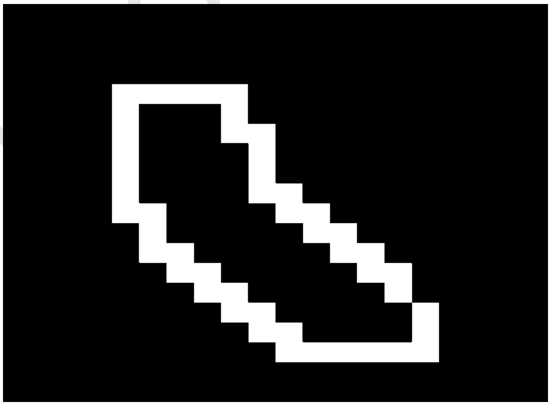


Figure 10. Fractal propagation $P_{\text{fake}} = 0.2$ edge detection.

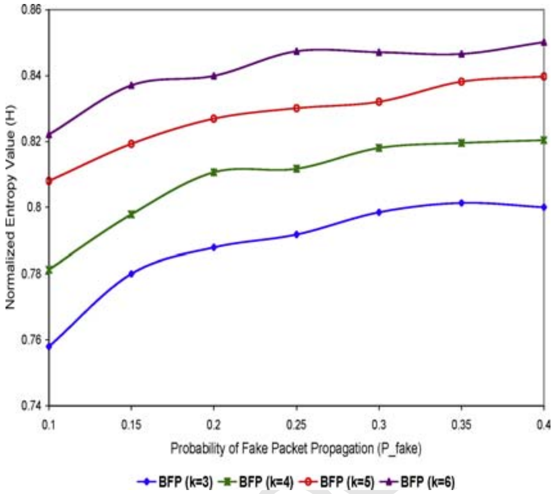


Figure 11. Fractal propagation—normalized entropy.

6.3. SECLOUD

Here we test SECLOUD under similar circumstances as the RW and the FP schemes. First we consider a single delegated source (labeled as S') and single delegated destination (labeled as D'). We use $k = 3$ and $B_S = B_D = 20$ for the simulation. The results of visualization of the traffic counts u_i at each node is shown in Figure 13. One can clearly see that as designed, SECLOUD creates two irregular clouds of identical broadcasting nodes (highlighted in grey cells) around the source and destination. This ensures the hiding of the source and destination among their cloud regions, since the source could be any node inside the cloud, (i.e., not necessarily at the center or the edge).

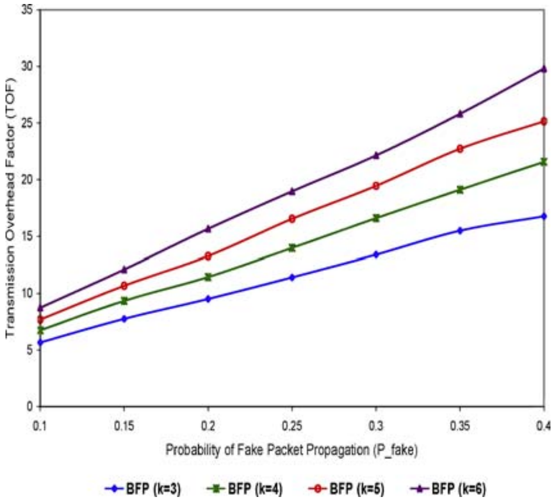


Figure 12. Fractal propagation—overhead.

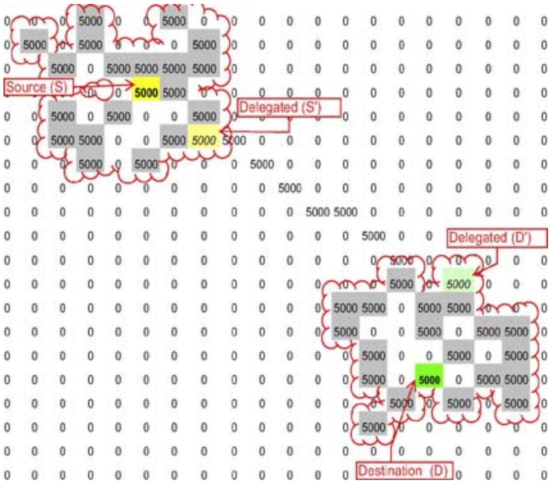


Figure 13. SECLOUD with single S-D.

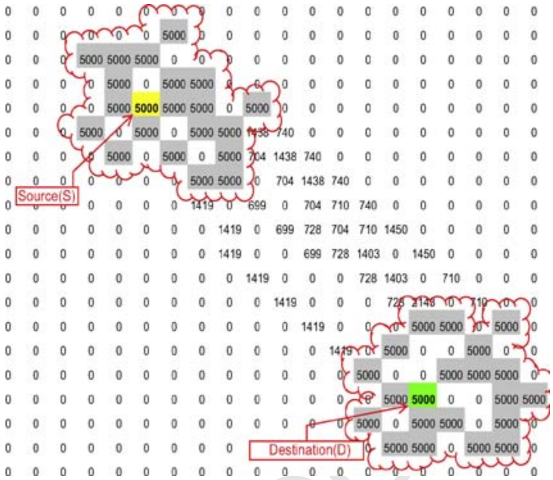


Figure 15. SECLOUD with multiple paths.

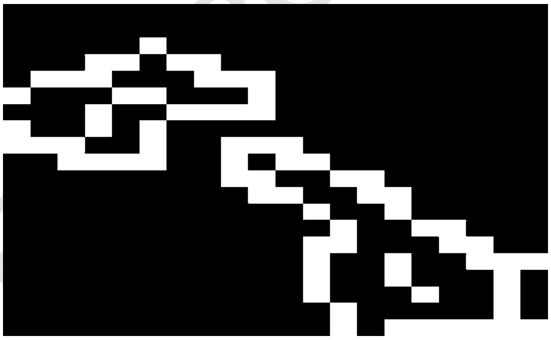


Figure 16. SECLOUD with single S-D: edge detection.

Figure 14 shows the 3-D visualization of the processed global traffic counts u_i of Figure 13. The attacker can see the communication path but no local or global maxima in the network. Thus, the linkability will not lead the attacker to the real source or destination, but only to the irregular cloud broadcasting region.

In Figure 15, we show a sample result for multiple delegated sources and delegated destinations. Figure 15 shows two irregular clouds of nodes broadcasting packets (highlighted in red) formed around the source and destination. The source sends 5000 packets using five delegated source–destination pairs for five transmission paths. Although using multiple paths is more complex to manage and setup, it has several advantages compared to a single path. In the case where an attacker is resident on a path, using several disjoint paths will allow some of the traffic to explicitly avoid that attacker. Moreover, using multiple paths will distribute the load of broadcasting the packets by intermediate nodes on the paths. The multiple paths also increase the unlinkability of the real source–destination pair. The anonymity level is similar to the case of SECLOUD with single path, as described previously.

The output of the edge detection algorithm for SECLOUD when a single delegated source and destination is used is shown in Figure 16. The shape extracted by the global attacker only reveals the cloud area and the path linking the two clouds.

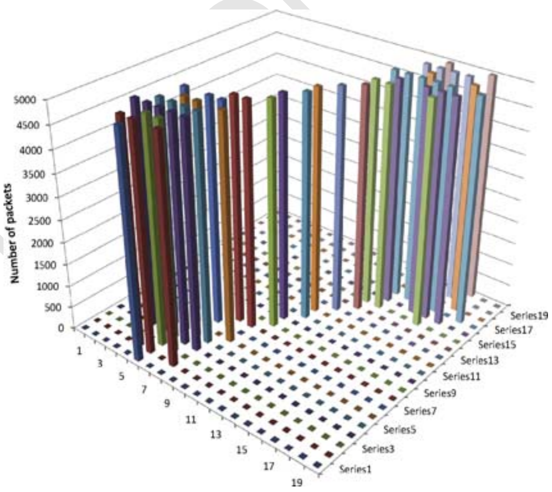


Figure 14. Three dimensional visualization of SECLOUD with single path.

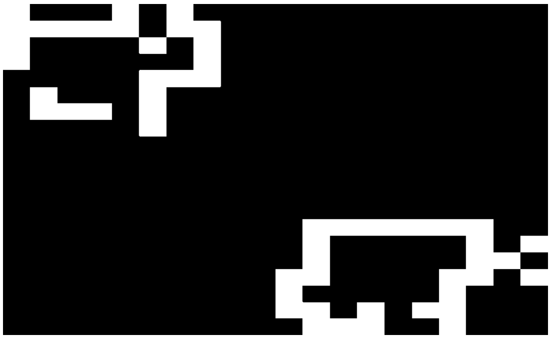


Figure 17. SECLOUD with multiple paths—edge detection.

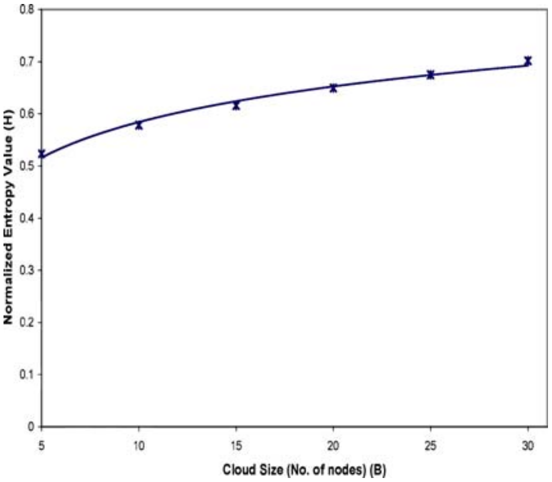


Figure 18. SECLOUD—normalized entropy.

Figure 17 shows the edge detection output when multiple delegated sources and destinations are used, resulting in multiple paths between the clouds. Notice that edge detection image analysis only shows the contour of the cloud regions not the paths.

In Figure 18, the impact of varying the cloud size $B_S = B_D = B$ on the average normalized network transmission entropy H is shown. The figure shows that the entropy slowly increases with the cloud size. Notice that the entropy values are low compared with the entropy values obtained from fractal propagation and random walk. However SECLOUD has higher anonymity and unlinkability and overall has a better ability to hide the source and destination nodes.

The overhead of the SECLOUD protocol *versus* the cloud size when both clouds are the same size (i.e., $B = B_S = B_D$) is shown in Figure 19. TOF here, as before, is the ratio of the total number of packet transmissions using SECLOUD to that with shortest path routing and no SECLOUD scheme.

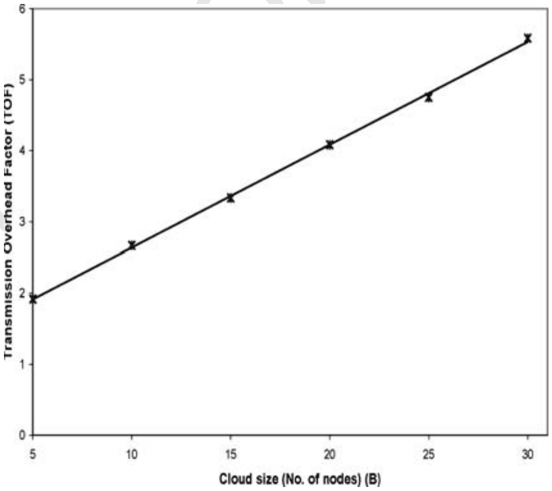


Figure 19. SECLOUD—transmission overhead.

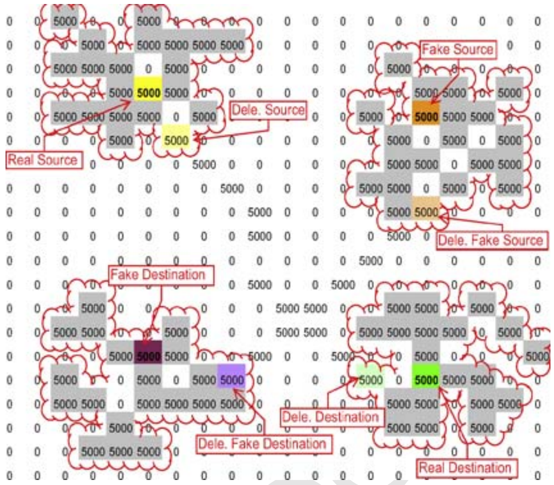


Figure 20. SECLOUD with single S-D and fake S-D.

Figure 19 shows a linear increase in TOF with larger cloud size. In comparison with RW one can see that with a cloud size $B = 20$, the TOF is about the same as the RW scheme with $(1 - P_r) = 0.3$. However, SECLOUD has a better ability to hide the source/destination than the RW scheme. Comparing SECLOUD with FP, one can see that the TOF of SECLOUD is much less than FP.

6.3.1. Fake clouds.

As noted earlier, one can further improve the obfuscation performance of SECLOUD by adding fake clouds. Figure 20 shows typical results for SECLOUD with a single delegated source destination pair with $B_S = B_D = 20$ and a fake source destination pair with a single delegated source destination pair and $B_S = B_D = 20$. The results can be compared to the case without a fake cloud in Figure 13. With the fake cloud, the attacker now not only needs to guess the location of the source and destination within a cloud, but

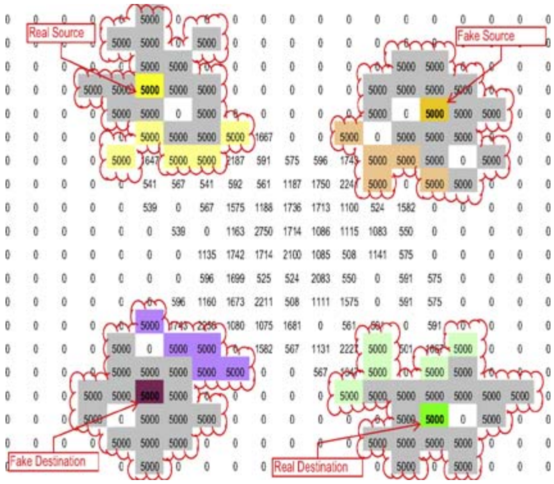


Figure 21. SECLOUD with multipaths and fake S-D.

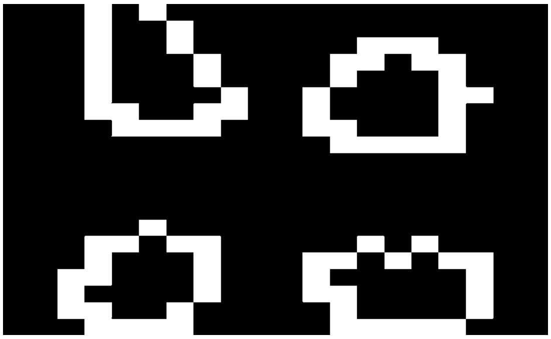


Figure 22. SECLOUD with multipaths and fake S-D.

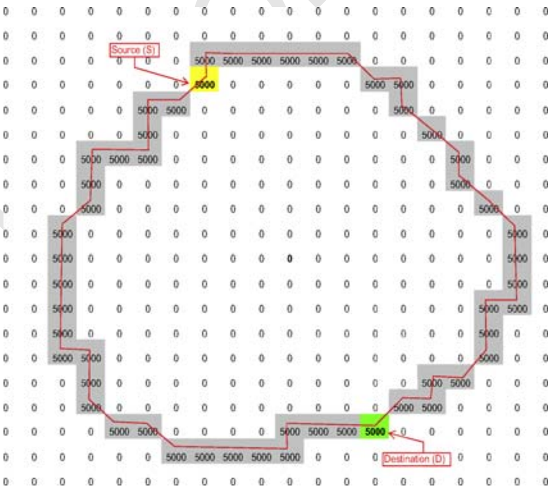
it also has to select the correct cloud to attack. Obviously, both the anonymity level and the unlinkability increase with the number of fake clouds and their sizes.

In Figure 21, we show a sample result with multiple delegated source destination pairs and a fake cloud with multiple delegated source destination pairs. In addition to having all the benefits of the previous approaches, this approach further increases the unlinkability and anonymity.

Figure 22 shows the corresponding edge detection results for the case of a single cloud and a fake cloud both with multiple delegated source and destinations. The edge detection image analysis shows the contour of the cloud regions only.

6.4. ANONYRING

Finally, we evaluate ANONYRING in the same fashion as RW, FP, and SECLOUD. In Figure 23, we show a typical visualization of traffic packet counts u_i for each node when the ANONYRING scheme is used. ANONYRING creates a ring of nodes that includes the source S and destination D nodes. The source node (S) transmits 5000 packets by randomly choosing the clockwise or counter clockwise direction along the ring to the destination node (D). In the example shown here the destination d is at a distance of 24 hops in the clockwise direction and 27 hops in the counter clockwise direction.



Source (S)
Destination (D)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	5000	5000	5000	5000	5000	5000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	5000	0	0	0	0	0	5000	5000	0	0	0	0	0
0	0	0	0	0	5000	5000	0	0	0	0	0	0	0	5000	0	0	0	0	0
0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	5000	0	0	0	0
0	0	0	5000	5000	5000	0	0	0	0	0	0	0	0	0	0	5000	0	0	0
0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	5000	0	0	0
0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0	0
0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0
0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0
0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0
0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	5000	0
0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0	0
0	0	5000	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0	0
0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	5000	5000	0	0	0
0	0	0	5000	0	0	0	0	0	0	0	0	0	0	5000	5000	0	0	0	0
0	0	0	0	5000	5000	0	0	0	0	5000	5000	5000	5000	0	0	0	0	0	0
0	0	0	0	0	0	5000	5000	5000	5000	5000	0	0	0	Destination (D)		0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 23. ANONYRING traffic shape.

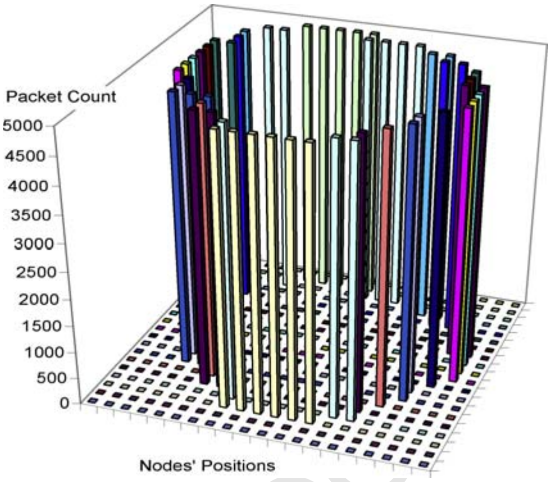


Figure 24. ANONYRING traffic shape—3-D visualization.

Figure 24 shows the 3-D traffic visualization chart of the packet count recorded at nodes in the ring only. One can see that the source and destination locations are well hidden and indistinguishable from the other nodes transmitting.

Figure 25 is the corresponding edge detection diagram of the ANONYRING technique that confirms the observations in Figures 23 and 24. The attacker will only be able to visualize the ring path not which pair of nodes are the true source and destination.

In Figure 26, the impact of varying the ring size R on the average normalized network transmission entropy H is shown. The entropy slowly increases with the ring size. Notice that the entropy values are slightly lower than SECLOUD and RW and much lower than the entropy values obtained from FP. However, ANONYRING has higher anonymity and unlinkability and overall hides the source and destination nodes better than RW and FP.

The overhead of the ANONYRING protocol *versus* the ring size is shown in Figure 27. TOF here is again the ratio of the total number of packet transmissions using ANONYRING to that from shortest path routing and no

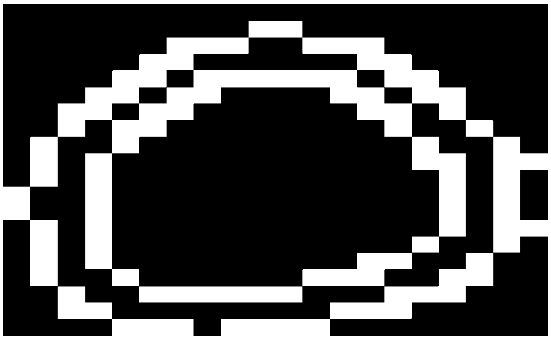


Figure 25. ANONYRING—edge detection.

Table I. Results summary.

Protocol	Anonymity	Unlinkability	Overhead
Random Walk	Least	Low	Medium
Fractal Propagation	Low	Low	Highest
SECLOUD	Variable with Cloud size can be High	High	Variable Low to High
ANONYRING	Variable with Ring size can be High	High	Moderate

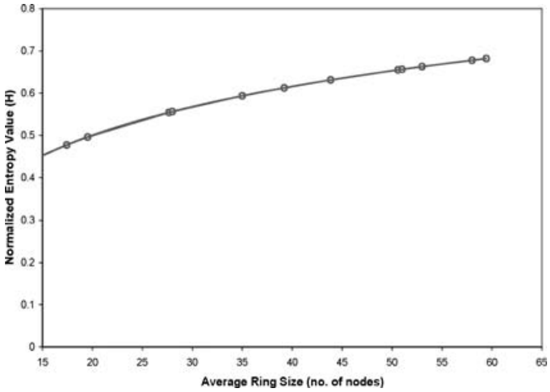


Figure 26. ANONYRING—normalized entropy.

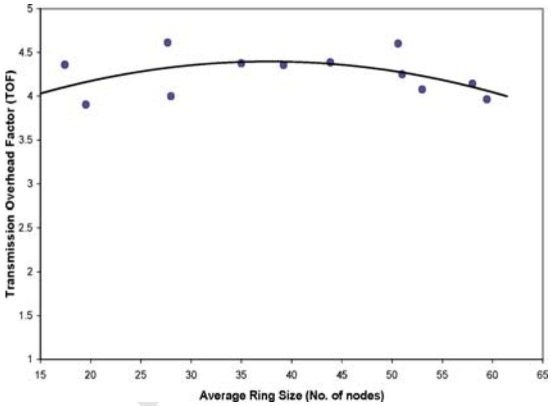


Figure 27. ANONYRING—transmission overhead.

ANONYRING in the network. In Figure 27, the dots represent the measured simulation results and the line is a curvelinear regression fit to the data. Note that the TOF in the figure does not vary much with ring size and actually goes down with larger ring sizes. This is not surprising given the procedure used in constructing Figure 27, specifically we picked a specific source–destination pair and then vary the ring size for four values, then we repeat the procedure for three different source–destination pairs. Thus, both RS and L increase resulting in a roughly constant TOF. In general for a fixed shortest path distance L between a communicating pair the TOF will increase linearly with increases in RS.

In comparison with RW one can see that the TOF is for small rings slightly higher than RW, but is lower for larger ring sizes. When compared with FP, ANONYRING has a much lower TOF than the FP scheme. Lastly when comparing SECLOUD with ANONYRING one can see that the TOF of ANONYRING is slightly less than SECLOUD. The anonymity of ANONYRING is higher than SECLOUD if the ring size RS is equivalent to SECLOUD cloud B_S, B_D sizes, specifically RS $= B_S + B_D$.

6.5. Summary of results and comparison

In comparing the four obfuscation schemes based on the sample results above and additional results not included here for the sake of brevity, we can make several interesting observations. First, all four schemes show that one must increase the overhead to increase the anonymity and unlinkability of a source destination pair. Secondly, entropy as a metric is misleading as to the actual anonymity and unlinkability provided, thus we do not consider it in the following. In comparing the four schemes we summarize our results in Table I. The random walk scheme while easily implemented and with moderate overhead was shown to provide little anonymity and unlinkability in the presence of an attacker with global traffic knowledge. Similarly, the fractal propagation scheme was found to have poor anonymity and unlinkability when the attacker can visualize the network wide traffic. Furthermore, the fractal propagation scheme had the highest overhead of any scheme studied. The SECLOUD scheme was shown to provide better anonymity and unlinkability and to have easily controlled parameters in the cloud sizes that directly effect the anonymity. Additional features such as increasing the number of delegated source destination pairs in the cloud and fake clouds can be added to increase anonymity. The overhead was found to be variable with these parameter but could be selected to compare favorably with random walk while providing much higher levels of anonymity and unlinkability. The ANONYRING scheme was also shown to provide the highest levels of anonymity and unlinkability with moderate levels of overhead and to have a single tunable parameter in the ring size RS.

The overhead performance and anonymity level of several cases of SECLOUD from simulations are shown in Table I. Using fake source–destination pairs will approximately double the overhead compared to the case without using any fake source–destination. As we increase the size of the cloud, the anonymity will increase but the overhead also will increase.

7. CONCLUSIONS

In this paper we addressed the problem of source–destination obfuscation in a wireless *ad hoc* network in the presence of an attacker with network-wide traffic knowledge. Previously, in the literature the random walk and fractal propagation schemes have been proposed to address the location privacy of source and destination nodes in *ad hoc* or sensor networks. Entropy of packet transmissions has been used as the main metric for evaluation. In this paper we showed that these current approaches can not provide sufficient obfuscation of the source, destination, and communication paths when the attacker can visualize the transmissions and infer contextual information. Furthermore, we showed that entropy is not a good metric for evaluating the obfuscation provided. To overcome the weaknesses of existing schemes, we proposed two novel techniques (1) SECLOUD: Source and Destination Seclusion using Clouds to obfuscate the true source/destination nodes and make them indistinguishable among a group of neighbor nodes, and (2) ANONYRING: Anonymous Ring which hides the source/destination nodes within a group of nodes that form a ring. Our comparative simulation study showed that both proposed techniques work well even under network-wide traffic visualization and analysis by an attacker. Furthermore the proposed techniques are shown to be superior to existing random walk and fractal propagation schemes in terms of overhead, anonymity, and unlinkability.

ACKNOWLEDGMENTS

This work was funded in part by the Army Research Office MURI grant W911NF-07-1-0318. The authors thank the anonymous reviewers for their comments to improve the paper.

REFERENCES

1. Nezhad AA, Miri A, Makrakis D. Location privacy and anonymity preserving routing for wireless sensor networks. *Computer Network* 2008; **52**(18): 3433–3452.
2. Mehta K, Liu D, Wright M. Location privacy in sensor networks against a global eavesdropper. *Proceedings of IEEE ICNP*, 2007.
3. Deng J, Han R, Mishra S. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. *Elsevier Journal of Pervasive and Mobile Computing on Security in Wireless Mobile Computing Systems* 2006; **2**(2): 159–186.
4. Yang Y, Shao M, Zhu S, Urgaonkar B, Cao G. Towards event source unobservability with minimum network traffic in sensor networks. *Proceedings of ACM WiSec*, 2008.
5. Jian Y, Chen S, Zhang Z, Zhang L. A novel scheme for protecting receiver's location privacy in wireless sensor networks. *IEEE Transactions on Wireless Comm.* 2008; **7**(10): 3769–3779.
6. Kamat P, Zhang Y, Trappe W, Ozturk C. Enhancing source-location privacy in sensor network routing. *Proceedings of IEEE ICDCS*, 2005.
7. Xi Y, Schwiebert L, Shi W. Preserving source location privacy in monitoring based wireless sensor networks. *Proceedings of SSN*, 2006.
8. Shao M, Yang Y, Zhu S, Cao G. Towards statistically strong source anonymity for sensor networks. *Proceedings of IEEE INFOCOM*, 2008.
9. Ozturk C, Zhang Y, Trappe W, Ott M. Source-location privacy for networks of energy-constrained sensors. *Proceedings of IEEE SEUS*, 2004.
10. Hoh B, Gruteser M. Protecting location privacy through path confusion. *Proceedings of SECURECOMM*, 2005.
11. Gruteser M, Grunwald D. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. *Mobile Networks Applied* 2005; **10**(3): 315–325.
12. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. *Proceedings of IEEE SSP*, 2003.
13. Du W, Deng J, Han YS, Varshney PK. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Transactions on Dependable and Secure Computer* 2006; **3**(1): 62–77.
14. Zhang Y, Liu W, Lou W. Anonymous communications in mobile ad hoc networks. *Proceedings of IEEE INFOCOM*, 2005.
15. Kong J, Hong X, Gerla M. An identity-free and on-demand routing scheme against anonymity threats in mobile ad hoc networks. *IEEE Transactions on Mobile and Computers* 2007; **6**(8): 888–902.
16. Shokri R, Yabandeh M, Yazdani N. Anonymous routing in manet using random identifiers. *Proc. of the Sixth International Conference on Networking*, 2007.
17. Lu R, Cao Z, Wang L, Sun C. A secure anonymous routing protocol with authenticated key exchange for ad hoc networks. *Computer Standard Interfaces* 2007; **29**(5): 521–527.
18. Boukerche A, El-Khatib K, Xu L, Korba L. Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. *Proceedings of IEEE LCN*, 2004.
19. Huang D. Traffic analysis based unlinkability measure for ieee 802.11b-based communication systems. *Proceedings of ACM WiSe*, 2006.
20. Grover W. *Mesh-based Survivable Networks: Options and Strategies for Optical , MPLS, and ATM Networking*. Prentice Hall PTR: 2003.[Q3]
21. Cahn RS. *Wide Area Network Design*. Morgan Kaufmann: 1998.[Q3]

22. Kuhn F, Zollinger A. Ad-hoc networks beyond unit disk graphs. *Proceedings of the Fifth International Workshop on Foundations of Mobile Computing*, 2003.

23. Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1986; **8**:679–714.

Author Query Form (SEC/220)

Special Instruction: Author please include responses to queries with your other corrections and return by e-mail.

Q1: Author: Please check the suitability of the suggested short title.
Q2: Author: Please check the presentation of correspondence details.
Q3: Author: Please provide the publisher location.
Q4: Author: Please provide a suitable figure (abstract diagram or illustration selected from the manuscript or an additional 'eye-catching' figure) and a short 'GTOC' abstract (maximum 80 words or 3 sentences) summarizing the key findings presented in the paper for Table of Content (TOC) entry.