SECLOUD: Source and Destination Seclusion Using Clouds for Wireless Ad hoc Networks

Razvi Doomun, Thaier Hayajneh, Prashant Krishnamurthy, and David Tipper
University of Pittsburgh
Pittsburgh, PA, USA
Emails: {mrazvi, hayajneh, prashant, dtipper} @sis.pitt.edu

Abstract—**The privacy of communicating entities in wireless ad hoc networks is extremely important due to the potential of their being identified and subsequently subjected to attacks (e.g., in military networks). Previously, the random walk and fractal propagation schemes have been proposed to address privacy of source and destination nodes in ad hoc or sensor networks. Entropy of packet transmissions has been used as the metric for comparison. In this paper, we show that under a global attacker that can eavesdrop on the overall data transmissions and count them, neither of these approaches provide sufficient privacy when the attacker can visualize the transmissions and infer contextual information. Moreover, we show that the entropy is not a useful metric in such a case. We propose SECLOUD: Source and Destination Seclusion using Clouds to obfuscate the true source/destination nodes and make them indistinguishable among a group of neighbor nodes which works well even under network-wide traffic visualization by a global attacker.**

I. Introduction

Malicious traffic analysis and privacy attacks against source and destination nodes in ad hoc networks are passive and difficult to detect in large wireless ad hoc networks [1]. Moreover, the disclosure of contextual information about network traffic patterns can be devastating in privacy-sensitive application scenarios. For example, in a military ad hoc network in the battlefield, 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 more critical to hide the location of the source as well as ensure the privacy of destination for quasi-stationary nodes in ad hoc networks. Spreading the network traffic among several paths is also a first step countermeasure against traffic analysis attacks. Different methods have been proposed in the literature for defending against traffic analysis and location privacy related attacks [2]–[5]. We note here that most of the research work assumes that all packets are encrypted link-by-link, padded to hide potential packet types through size, and use anonymous routing schemes to avoid detection of routes during route set-up. Further, energy constraints and overhead are typically not factored because obfuscation requires transmission of dummy packets (discussed later).

However, most existing solutions to protect privacy of source or destination nodes do not consider the possibility of attackers having a complete view of the network traffic, in which case they fail to work well. In this paper we address the problem of secluding source/destination nodes in the presence of an attacker with complete view of the network topology and traffic. In wireless ad hoc networks, 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 it does not know the overall network topology, whereas a global attacker is able to visualize the overall network topology and is capable of network-wide traffic rate monitoring and time-correlation attacks. Network-wide rate monitoring attack involves counting the number of transmitted/received packets around every node in the network, while time-correlation attacks involve finding the communication patterns by analyzing latencies between packet transmissions around nodes in the network.

In this work, we propose a simple and efficient technique for source/destination privacy against such a global adversary. The proposed protocol, SECLOUD: Source and Destination Seclusion using Clouds, hides the source and destination nodes in a group of nodes (called "cloud") that are indistinguishable. The performance of SECLOUD is evaluated assuming a global attacker model capable of using network-wide traffic visualization. We compare SECLOUD with two commonly used privacy techniques, Random Walk and Fractal Propagation [2], [6]. SECLOUD provides better privacy than either scheme in the presence of a global traffic visualization attack. We show that entropy, which is a common privacy metric, is not always an adequate measure of anonymity strength and also quantify the privacy level using anonymity and unlinkability metrics. Our simulations show that SECLOUD has a lower overhead compared to the fractal propagation and random walk schemes.

The rest of the paper is organized as follows. Section 2 describes related work on popular privacy techniques. The attacker model, network assumptions, and privacy metrics are presented in Section 3. The details of SECLOUD protocol are explained in Section 4. Section 5 presents the simulation set-up and results discussion. Section 6 concludes the paper.

II. Related Work

Previous works on privacy mechanisms for ad hoc wireless networks 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 deal with location privacy in wireless sensor networks where the entity to be protected is the sink node [2], [5] or the sensing sources only [6]–[8].

Random Walk: Kamat et al. [7] and Ozturk et al. [9] proposed similar techniques based on a random walk approach for source location privacy in sensor networks against an external local adversary. Xi et al. [6] developed an improved random walk approach that starts from both source and sink. The random walk operates by forwarding packets probabilistically from the source through an unpredictable route (with extra hops than the shortest path) to reach the destination [6]. Using erratic per-packet route is very effective in resisting rate monitoring and traffic analysis attacks by adversaries monitoring a limited set of nodes. The random walk increases the time to trace the source location and thus the source can communicate privately with the destination for a longer time. However, it is hard to completely obfuscate the general trend of the traffic flow towards the destination location.

Fractal Propagation: Deng et al. [2] have proposed the fractal progation technique to counteract local rate monitoring and correlation attacks against location privacy in sensor networks. Fractal propagation overcomes one of the drawbacks of the random walk scheme by introducing fake packet spreading to combat time correlation attacks [2]. When a neighbor node overhears a packet transmission, it probabilistically generates a fake packet with k-hop lifetime and forwards it to one of its neighbors. 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 diffused than the random walk method. However, fractal propagation cannot totally obfuscate traffic patterns near the source and destination area, because all real packets diverge from the source node and converge towards the destination node. Thus, eavesdropping by a global attacker which can visualize the network-wide traffic can expose the source and destination regions as we show later.

In [10], a path confusion algorithm is used to increase source location anonymity against a local adversary model. Mehta et al. in [3] proposed a privacy scheme under a global external attack model by hiding the real source among $k-1$ fake sources simulating the mobility pattern of real sources in sensor network. In [8], under the global attack model, the authors proposed statistically strong source anonymity by employing network-wide dummy messages to achieve global privacy. In [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 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 and number of fake sources are important factors that determine the effectiveness of the aforementioned privacy mechanisms.

III. Models and Assumptions

A. Network model

In this work, we consider an ad hoc network with nodes being distributed in a grid like manner (or randomly). Our assumptions are very similar to most related work in the literature. The IEEE 802.11 standard for physical and MAC layers is assumed and nodes do not use RTS or CTS to avoid revealing communication peers. 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 will also be hidden and not advertised. In [11] the authors used short-lived disposable MAC addresses to prevent the real node IDs from being revealed to adversaries. A similar technique will be assumed in this paper to avoid any identification of nodes. We assume the existence of a key management protocol that can distribute pair-wise keys between nodes or public-private key pairs for each node [12], [13]. Any of these schemes can be used to set up pairwise keys and authenticate nodes' relationship, but we omit such details here in this paper. 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 [14]–[18]. An anonymous routing protocol allows neighbor nodes to authenticate each other without revealing their identities. For example, in [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 employ pairwise *shared link identifiers* between neighbor nodes which are created and established during neighborhood authentication.

B. Attack Model

An external, global, and powerful attacker model is assumed in this paper that has combined capabilities of different existing attack models as described in [2], [3], [19]. The attacker has complete knowledge of 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 attacks for all traffic in the network (which is a stronger attack than corresponding ones assumed elsewhere). The attacker will visualize all transmitted/received packets in the network and determine the traffic density on every link in the network. However, the attacker is passive and cannot compromise nodes in the network. A possible method for this attack is by deploying an overlay network with several malicious nodes simply to sense traffic from the given ad hoc network, similar to the idea in [3]. These nodes can collect information and collaborate with a centralized entity using a different band. We investigate the ability of privacy techniques to withstand this powerful global attacker. We consider a single source and destination in the network that makes it easiest for an attacker to identify them, but SECLOUD can be deployed with multiple sources and destinations as well.

C. Privacy Evaluation Metrics

We analyze the performance of SECLOUD, random walk and fractal propagation using the metrics – anonymity, unlinkability [19], and entropy.

1) Anonymity: 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 nodes that have the same behavior, then the level of anonymity is $\lambda = 1 - (1/A)$. If the attacker estimates that the source-destination pair is in the anonymity sets A_1, A_2,... A_n, where

$|A|$ is the size of a set A and n is number of sets, then, the anonymity level for this node of interest is given by $\lambda = 1 - \{1/|A_1 \cup A_2 \cup ... \cup A_n\}|$. Thus, the anonymity level of a node (source/destination in our case) depends on the number of nodes in the anonymous zone, which is the cloud area in SECLOUD.

2) Unlinkability: We employ 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. More powerful edge detection algorithms are also possible (we show one example). Details are provided in Section V.

3) Entropy: Entropy, a common metric for quantifying uncertainty in information theory, is often used as a privacy measure [2]. It is a scalar number that does not capture the amount of contextual information, such as paths analysis, that disclose hints about the location of source and destination nodes.

IV. 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 next.

First, the source node S broadcasts a hello message to discover all its one-hop neighbors $N(1, i)$ for $i = 1, 2, ..m$, where m is the total number of neighbor nodes. Then, the nodes in $N(1, i)$ discover their respective neighbors $N(2, i)$ which are two-hops away from node S. Consequently, source node S constructs the list: $N(1, i), N(2, i), N(3, i)...N(k, i)$, where $N(k, i)$ is the set of k^{th} hop neighbors of node S. This initialization process of neighbor discovery is done periodically by all nodes in the network. This will ensure that the attacker cannot determine which of the nodes performing the initialization will be the source.

Let the cloud region be of maximum width k hops from the source. For e.g, with $k = 3$, source node S will randomly select a number of nodes, B, such that $B \subseteq \{N(1, i) \cup N(2, i) \cup N(3, i)\}$. This enables reducing overhead compared to schemes that use dummy transmissions everywhere or by every node and keep the 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 and to forward real packets when available from source to delegated sources (see next). Dummy packets are simply dropped. The destination node D which also does the same initialization procedure will construct a cloud. *Note the size of the source and the destination clouds can be different by using different values of k for each, depending on the privacy strength needed on each side.*

Node S randomly selects one or more nodes from the set B to act as *delegated sources*. Similarly D randomly selects one or more nodes in its cloud to act as *delegated destinations*. We consider the case of one delegated source/destination as a special case in Section V. Delegated sources find routes to delegated destinations using a suitable anonymous routing protocol (e.g., see [14]). In any case, local broadcasts/relays from S to the delegated sources and from the delegated destinations to D ensure delivery of data locally. Nodes in B are picked such that there is connectivity between S and the delegated sources. A similar strategy is employed in the destination cloud.

With the above set-up, SECLOUD achieves source/destination privacy in a local regions. In single source-destination scenario, the global attacker will have to guess the source or the destination from the set of nodes B which is the cloud size. To improve privacy, 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 S-D clouds and will communicate with each other. In the best case, paths between fake source/destination clouds will intersect the real traffic flow. Fake clouds increase the likelihood of misleading the attacker who may attack (e.g., by jamming) the fake regions instead of the real regions. When there are multiple sources and destinations with different traffic paths crossing each other in the same time period, fake clouds may be omitted. Nodes in the real source-destination clouds and all the nodes in the fake source-destination regions will need to be loosely synchronized so that the global attacker observes packet transmissions in the same time period.

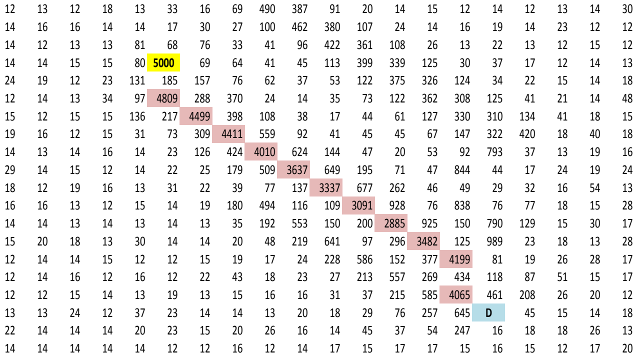
V. Evaluation

We simulated a large network of 400 nodes distributed in an area of 2000m × 2000m with average node degree between 7 and 8. We employed the Quasi-Unit disk graph (Q-UDG) connectivity [20] with some percentage of perturbations and random distribution introduced for grid network topologies. The coordinates of each node x and y were randomly chosen using uniform random variables in the ranges $(x - px_i, x + px_i)$ and $(y - py_i, y + py_i)$, respectively, where p is the perturbation parameter and x_i and y_i are the spacing between the nodes in the x and y directions, respectively ($p = 0.2$ and $x_i = y_i = m \times 100$ for integral m were assigned for all our simulations). For random node distribution, the coordinates of the nodes x and y were randomly and independently chosen in the range from 0 to 2000 m. After the nodes are distributed, the Q-UDG connectivity model and transmission range is used to determine the network topology. 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 ($0 \leq \alpha \leq 1$). In our simulation we set $\alpha = 0.2$. 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 $(R-d)/(R-\alpha R)$. The source node is randomly chosen from the region ($100m \leq$x$\leq 900m$) and ($100m \leq$y$\leq 900m$) and destination node is randomly selected from ($1100m \leq$x$\leq 1900m$) and ($1100m \leq$y$\leq 1900m$)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.

Metrics: If node i transmits u_i packets and a total of V packets were transmitted in the network in time 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 determine unlinkability and anonymity as follows. To visualize linkable communications, the attacker can simply plot a 3-D graph of the number of packets transmitted by each node in time T with the approximate location of each node. A second approach is to convert this information into an image and use an edge detection algorithm to reveal source/destination locations as well as the communication route. We show an example of the Canny edge detection algorithm [21], which is a popular image processing approach in computer vision. 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. Finally, to better quantify linkability, the attacker will sample n of the nodes that have the highest number of packets transmitted in 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 graph of nodes, the number of packets transmitted and the **marked nodes** is 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. In the case of random walk 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.

A. Random Walk Technique

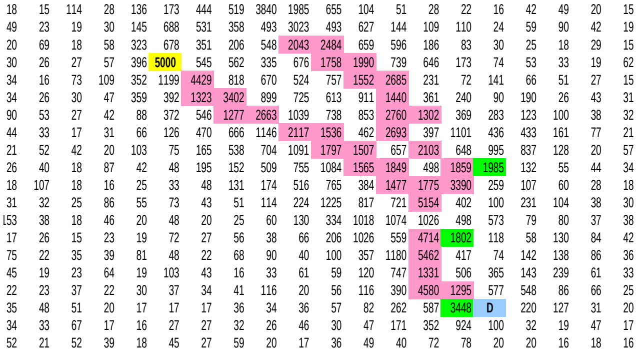
With random walk, [2], [7], packets are forwarded in a random fashion from one hop to the next until they reach the destination. The route taken by each packet from source to destination is unpredictable. A probability $(0.5 \leq P_r \leq 1.0)$ is used at each hop to decide how random the forwarding is. If $P_r = 1.0$, there is no randomness and the packet is sent to next hop node on the shortest path to the destination.



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	96	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

Fig. 1. Random Walk $P_r = 0.9$ – marking nodes

In Fig. 1, a matrix of the number of packets u_i transmitted by each node i is shown for random walk with $Pr = 0.9$ (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 in Fig. 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.



18	15	114	28	136	173	444	519	3840	1985	655	104	51	28	22	16	42	49	20	15
49	23	19	30	145	688	531	358	493	3023	493	627	144	109	110	24	59	90	42	19
20	69	18	58	323	678	351	206	548	2043	2484	659	596	186	83	30	25	18	29	15
30	26	27	57	396	**5000**	545	562	335	676	1758	1990	739	646	173	74	53	33	19	62
34	16	73	109	352	1199	4429	818	670	524	757	1552	2685	231	72	141	66	51	27	15
34	26	30	47	359	392	1323	3402	899	725	613	911	1440	361	240	90	190	26	43	31
90	53	27	42	88	372	546	1277	2663	1039	738	853	2760	1302	369	283	123	100	38	32
44	33	17	31	66	126	470	666	1146	2117	1536	462	2693	397	1101	436	433	161	77	21
21	52	42	20	103	75	165	538	704	1091	1797	1507	657	2103	648	995	837	128	20	57
26	40	18	87	42	48	195	152	509	755	1084	1565	1849	498	1859	1985	132	55	44	34
18	107	18	16	25	33	48	131	174	516	765	384	1477	1775	3390	259	107	60	28	18
31	32	25	86	55	73	43	51	114	224	1225	817	721	5154	402	100	231	104	38	30
153	38	18	46	20	48	20	25	60	130	334	1018	1074	1026	498	573	79	80	37	38
17	26	15	23	19	72	27	56	38	66	206	1026	559	4714	1802	118	58	130	84	42
75	22	35	39	81	48	22	68	90	40	100	357	1180	5462	417	74	142	138	86	36
45	19	23	64	19	103	43	16	33	61	59	120	747	1331	506	365	143	239	61	33
22	23	37	22	30	37	34	41	116	20	56	116	390	4580	1295	577	548	86	66	25
35	48	51	20	17	17	17	36	34	36	57	82	262	587	3448	**D**	220	127	31	20
34	33	67	17	16	27	27	32	26	46	30	47	171	352	924	100	32	19	47	17
52	21	52	39	18	45	27	59	20	17	36	49	40	72	78	20	20	16	18	16

Fig. 2. Random Walk $P_r = 0.7$ – marking nodes

In Fig. 2, we reduce P_r to 0.7 and repeat the process. In this case ($n = 10$, $\beta = 0.5$, $\beta U = 2130$), yields a fuzzy region. We reduced β to 0.3 resulting in $\beta U = 1278$ and a sharper set of paths is obtained as shown in Figures 2 and 4. The privacy is increased since more than one communication path appears to exist. The destination node can be guessed to be near three nodes with u_i's 1985, 1802 and 3448 and destination anonymity is 2/3 (i.e., $|A| = 3$). The source node packet count is still a global maxima in its area and can still be detected.

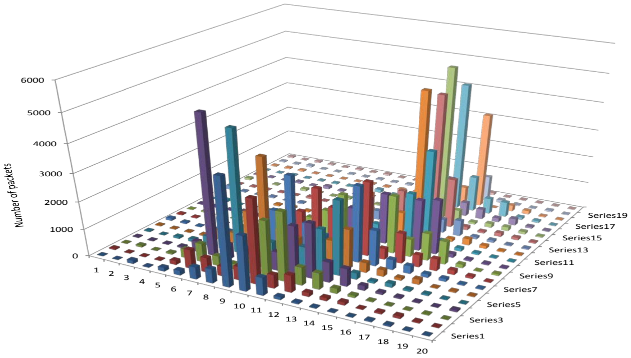


Fig. 3. Random Walk $P_r = 0.7$ – 3-D graph of u_i

The 3D traffic graph for this case is shown in Fig. 3. The output of the edge detection algorithm is shown in Fig. 4. The same conclusions can be extracted by the global attacker from these figures. We omit some of these graphs for subsequent privacy techniques (fractal propagation and SECLOUD).

The overhead cost with random walk (average from 15 simulations) is shown in Table I for different values of P_r. The average path length to reach the destination and the corresponding computed entropy values are also shown in the same table. The transmission overhead (TO) is the ratio of total number of packets transmitted that with the shortest single

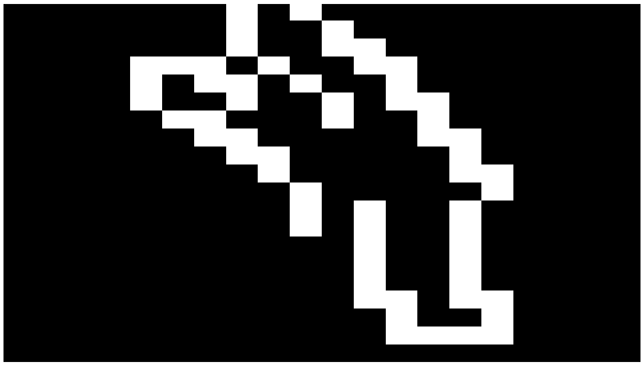


Fig. 4. Random Walk $P_r = 0.7$ – Edge Detection

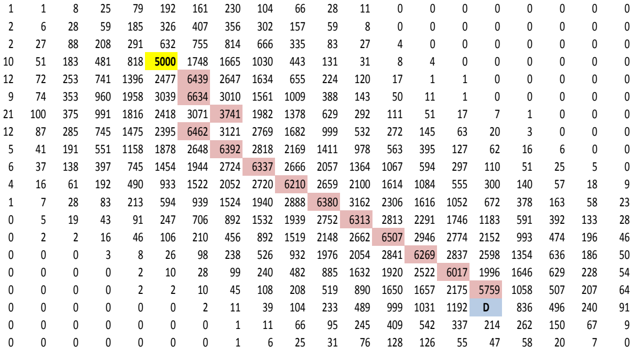
$(1 - P_r)$	Av. path length	TO	Mean Entropy
0.5	60.6	4.66	7.222
0.4	39.5	3.04	7.266
0.3	30.4	2.34	6.853
0.2	25.7	1.98	6.854
0.1	15.0	1.15	6.262

TABLE I
RW PROTOCOL

path transmission from a fixed source to a fixed destination 13 hops away. The transmission overhead can be as high as 466 % with for $P_r = 0.5$. Even with $P_r = 0.7$, the overhead is 234% and the anonymity and unlinkability is not adequate. Although, the entropy values increased with decreasing P_r, they provide little information about the achieved privacy.

B. 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 with extra overhead.



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	356	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	456	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

Fig. 5. Fractal Propagation $P_{fake} = 0.1, K = 5$ – marking nodes

The results of traffic visualization for one simulation sample by marking nodes ($n = 10$, $\beta = 0.5$, $\beta U = 3197$) is shown in Fig. 5 for $P_{fake} = 0.1, K = 5$. 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 3D traffic graph shown in Fig. 6, 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 .

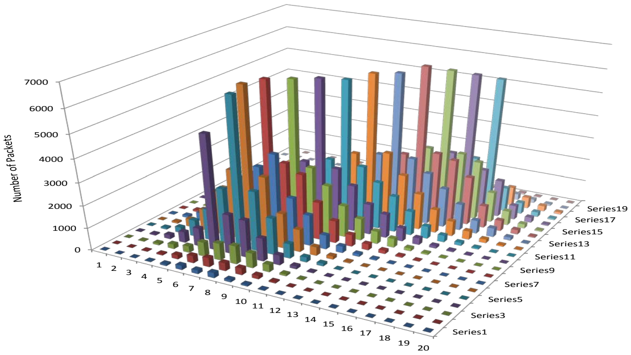
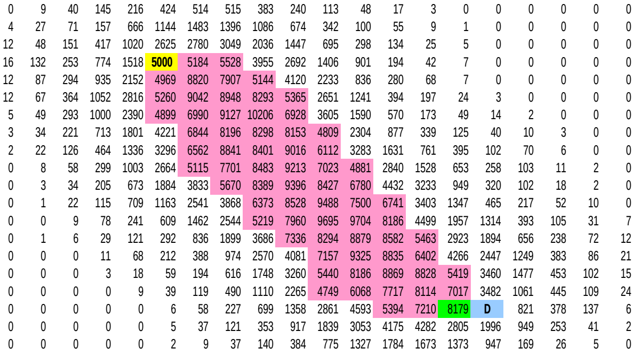


Fig. 6. Fractal Propagation $P_{fake} = 0.1, K = 5$ – 3-D graph



0	9	40	145	216	424	514	515	383	240	113	48	17	3	0	0	0	0	0	0
4	27	71	157	666	1144	1483	1396	1086	674	342	100	55	9	1	0	0	0	0	0
12	48	151	417	1020	2625	2780	3049	2036	1447	695	298	134	25	5	0	0	0	0	0
16	132	253	774	1518	**5000**	5184	5528	3955	2692	1406	901	194	42	7	0	0	0	0	0
12	87	294	935	2152	4969	8820	7907	5144	4120	2233	836	280	68	7	0	0	0	0	0
12	67	364	1052	2816	5260	9042	8948	8293	5365	2651	1241	394	197	24	3	0	0	0	0
5	49	293	1000	2390	4899	6990	9127	10206	6928	3605	1590	570	173	49	14	2	0	0	0
3	34	221	713	1801	4221	6844	8196	8298	8153	4809	2304	877	339	125	40	10	3	0	0
2	22	126	464	1336	3296	6562	8841	8401	9016	6112	3283	1631	761	395	102	70	6	0	0
0	8	58	299	1003	2664	5115	7701	8483	9213	7023	4881	2840	1528	653	258	103	11	2	0
0	3	34	205	673	1884	3833	5670	8389	9396	8427	6780	4432	3233	949	320	102	18	2	0
0	1	22	115	709	1163	2541	3868	6373	8528	9488	7500	6741	3403	1347	465	217	52	10	0
0	0	9	78	241	609	1462	2544	5219	7960	9695	9704	8186	4499	1957	1314	393	105	31	7
0	1	6	29	121	292	836	1899	3686	7336	8294	8879	8582	5463	2923	1894	656	238	72	12
0	0	0	11	68	212	388	974	2570	4081	7157	9325	8835	6402	4266	2447	1249	383	86	21
0	0	0	3	18	59	194	616	1748	3260	5440	8186	8869	8828	5419	3460	1477	453	102	15
0	0	0	0	9	39	119	490	1110	2265	4749	6068	7717	8114	7017	3482	1061	445	109	24
0	0	0	0	0	6	58	227	699	1358	2861	4593	5394	7210	8179	**D**	821	378	137	6
0	0	0	0	0	5	37	121	353	917	1839	3053	4175	4282	2805	1996	949	253	41	2
0	0	0	0	0	2	9	37	140	384	775	1327	1784	1673	1373	947	169	26	5	0

Fig. 7. Fractal Propagation $P_{fake} = 0.3$

In Fig. 7, for the same source and destination nodes, a higher probability of fake packet generation $(P_{fake} = 0.3)$ is used with fractal propagation. The source and destination nodes can both still be located using the same methodology (for $n = 10$, $\beta = 0.5$, in this case $\beta U = 4710$).

P_{fake} and K values	TO	# Fake Paths	Mean Entropy
$P_{fake} = 0.1, K = 5$	4.75	3750	6.996
$P_{fake} = 0.2, K = 5$	8.29	7289	7.143
$P_{fake} = 0.3, K = 5$	11.59	10591	7.269
$P_{fake} = 0.1, K = 6$	5.11	3425	7.127
$P_{fake} = 0.2, K = 6$	9.53	7111	7.266
$P_{fake} = 0.3, K = 6$	13.35	10287	7.323

TABLE II
BFP TRANSMISSION OVERHEAD

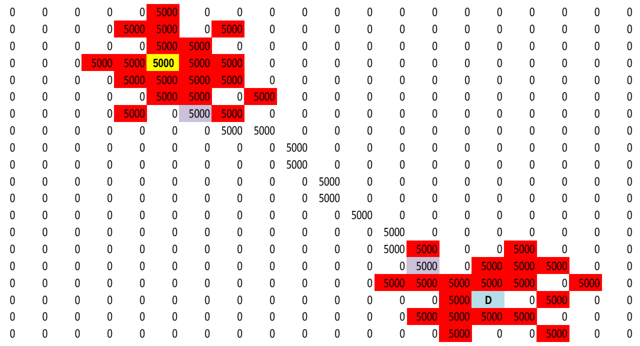
The overhead cost of the fractal propagation protocol is shown in Table II. The transmission overhead (TO) is again the ratio of total number of packet transmissions to that with shortest single path transmission from a fixed source to a fixed destination 13 hops away. The number of different fake paths produced by fake packet propagation and the mean entropy values are also shown in the same table. For the shortest single path case, the entropy computed is 3.985.

Obviously as P_{fake} is increased and K is increased, the overhead will be significantly increased. Even $P_{fake} = 0.3$ and $K = 5$ results in a high transmission overhead (TO) of 1159 %, but the anonymity and unlinkability is poor. The mean entropy values increase but provide no real information about privacy.

In comparison with fractal propagation, the unlinkability with random walk is better when P_r is decreased. As illustrated in Fig. 2 the series of local maxima nodes tend to be irregular which make it harder for the global attacker to follow the traffic flow precisely.

C. SECLOUD

Next we test SECLOUD under identical circumstances as random walk and fractal propagation, with and without fake clouds.



0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5000	5000	0	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5000	5000	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	5000	5000	5000	5000	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5000	5000	5000	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5000	5000	0	5000	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5000	0	5000	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	5000	5000	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	5000	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	5000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	5000	5000	0	0	5000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0	5000	5000	5000	0	0
0	0	0	0	0	0	0	0	0	0	0	0	5000	5000	5000	5000	5000	0	5000	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	D	0	5000	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	5000	5000	5000	5000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0	0	5000	0	0

Fig. 8. SECLOUD with single path – marked nodes

First we consider a single delegated source and single delegated destination (gray cells) in Fig. 8. A cloud size $B < 4k(k+1)$ was chosen in the case of grid networks where k is the selected random hop distance from the source/destination. With $k = 3$, we picked $B = 20$ for this simulation. As it was designed to perform, two irregular clouds of identical broadcasting nodes of 5000 packets (highlighted in red) are formed around the source and destination. This will ensure the location privacy 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. The probability for a global attacker to guess the location of the source will be 1/total number of nodes in both clouds. Note that the size of the clouds B can be adjusted by the SECLOUD protocol, which means that the anonymity can be controlled. The linkability will not lead the attacker to the real source or destination, but only to the irregular cloud broadcasting region.

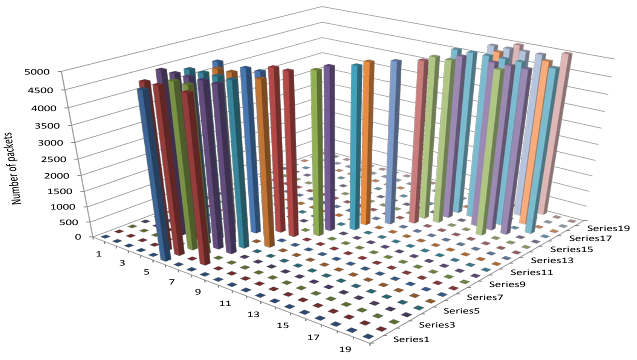
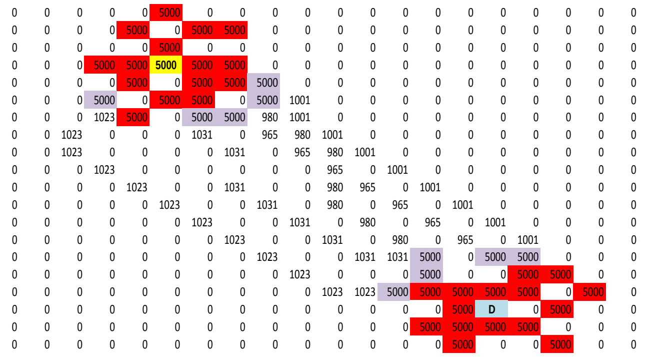


Fig. 9. SECLOUD with single path – 3-D Graph

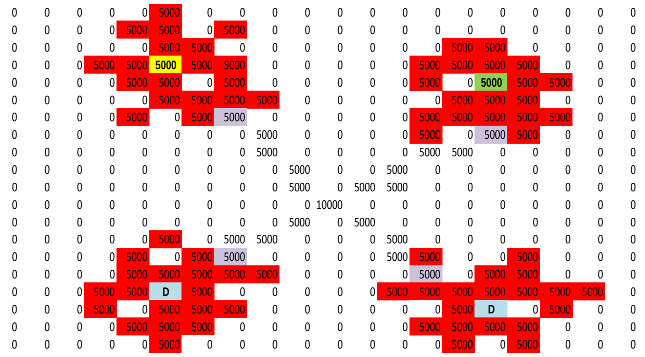
Fig. 9 shows no local or global maxima in the network and confirms from what we have concluded from Fig. 8.



0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5000	0	5000	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	5000	5000	5000	5000	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5000	0	5000	5000	5000	0	0	0	0	0	0	0	0	0	0	0
0	0	0	5000	0	5000	5000	0	5000	1001	0	0	0	0	0	0	0	0	0	0
0	0	0	1023	5000	0	5000	5000	980	1001	0	0	0	0	0	0	0	0	0	0
0	0	1023	0	0	0	1031	0	965	980	1001	0	0	0	0	0	0	0	0	0
0	0	1023	0	0	0	0	1031	0	965	980	1001	0	0	0	0	0	0	0	0
0	0	0	1023	0	0	0	0	0	0	965	0	1001	0	0	0	0	0	0	0
0	0	0	0	1023	0	0	1031	0	0	980	965	0	1001	0	0	0	0	0	0
0	0	0	0	0	1023	0	0	1031	0	980	0	965	0	1001	0	0	0	0	0
0	0	0	0	0	0	1023	0	0	1031	0	980	0	965	0	1001	0	0	0	0
0	0	0	0	0	0	0	1023	0	0	1031	0	980	0	965	0	1001	0	0	0
0	0	0	0	0	0	0	0	1023	0	0	1031	1031	5000	0	5000	5000	0	0	0
0	0	0	0	0	0	0	0	0	1023	0	0	0	5000	0	0	5000	5000	0	0
0	0	0	0	0	0	0	0	0	0	1023	1023	5000	5000	5000	5000	5000	0	5000	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	D	0	5000	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	5000	5000	5000	5000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5000	0	0	5000	0	0

Fig. 10. SECLOUD with multiple paths - marked nodes

In Fig. 10, we show a sample result for multiple delegated sources and delegated destinations. Fig. 10 again shows two irregular clouds of nodes broadcasting 5000 packets (highlighted in red) formed around the source-side and destination-side. The real source sends 5000 packets using five delegated source-destination pairs (highlighted in gray), which also mean there are 5 transmission paths. Although using multiple paths is more complex to manage and setup, it has two advantages compared to single path. In the case where an attacker is resident on a path, using several disjoint paths will explicitly avoid that attacker. Moreover, using multiple paths will distribute the load of broadcasting the packets by intermediate nodes on those 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.

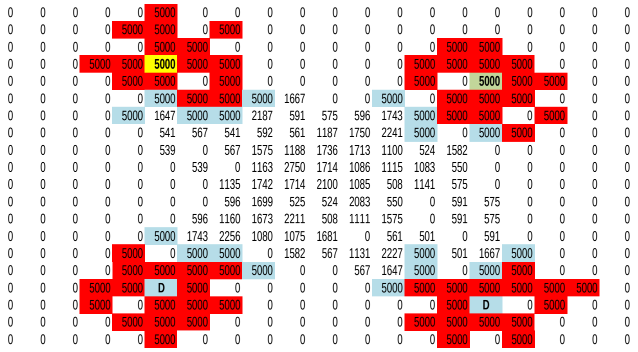


0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5000	5000	0	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5000	5000	0	0	0	0	0	0	0	5000	5000	0	0	0	0
0	0	0	5000	5000	5000	5000	5000	0	0	0	0	0	5000	5000	5000	5000	0	0	0
0	0	0	0	5000	5000	0	5000	0	0	0	0	0	5000	0	5000	5000	5000	0	0
0	0	0	0	0	5000	5000	5000	5000	0	0	0	0	0	5000	5000	5000	0	0	0
0	0	0	0	5000	0	5000	5000	0	0	0	0	0	5000	5000	5000	5000	5000	0	0
0	0	0	0	0	0	0	0	5000	0	0	0	0	5000	0	5000	5000	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	0	5000	0	0	5000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5000	0	5000	5000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	10000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5000	0	5000	0	0	0	0	0	0	0	0
0	0	0	0	0	5000	0	5000	5000	0	0	0	5000	0	0	0	0	0	0	0
0	0	0	0	5000	0	5000	5000	0	0	0	0	5000	5000	0	0	5000	0	0	0
0	0	0	0	5000	5000	5000	5000	5000	0	0	0	0	5000	0	5000	5000	0	0	0
0	0	0	5000	5000	D	5000	0	0	0	0	0	5000	5000	5000	5000	5000	5000	5000	0
0	0	0	5000	0	5000	5000	5000	0	0	0	0	0	0	5000	D	0	5000	0	0
0	0	0	0	5000	5000	5000	0	0	0	0	0	0	5000	5000	5000	5000	0	0	0
0	0	0	0	0	5000	0	0	0	0	0	0	0	0	5000	0	5000	0	0	0

Fig. 11. SECLOUD with single path and fake clouds

Fig. 11 shows sample results for SECLOUD (single path with a fake source-destination pair). Compared to the previous case without a fake cloud, the attacker now not only needs to guess the location of the source within a cloud, but it also has to select the correct cloud and the guess the real source within it. The anonymity level in this case is doubled with one fake source-destination, and the unlinkability of real source-destination is higher. Moreover, it is not clear which cloud is the source and which is the destination.

In Fig. 12, we show a sample result with multiple paths and one fake source-destination pair. In addition to having all



0	0	0	0	0	5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5000	5000	0	5000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5000	5000	0	0	0	0	0	0	0	5000	5000	0	0	0	0
0	0	0	5000	5000	5000	5000	5000	0	0	0	0	0	5000	5000	5000	5000	0	0	0
0	0	0	0	5000	5000	0	5000	0	0	0	0	0	5000	0	5000	5000	5000	0	0
0	0	0	0	0	5000	5000	5000	5000	1667	0	0	5000	0	5000	5000	5000	0	0	0
0	0	0	0	5000	1647	5000	5000	2187	591	575	596	1743	5000	5000	5000	0	5000	0	0
0	0	0	0	0	541	567	541	592	561	1187	1750	2241	5000	0	5000	5000	0	0	0
0	0	0	0	0	539	0	567	1575	1188	1736	1713	1100	524	1582	0	0	0	0	0
0	0	0	0	0	0	539	0	1163	2750	1714	1086	1115	1083	550	0	0	0	0	0
0	0	0	0	0	0	0	1135	1742	1714	2100	1085	508	1141	575	0	0	0	0	0
0	0	0	0	0	0	0	596	1699	525	524	2083	550	0	591	575	0	0	0	0
0	0	0	0	0	0	596	1160	1673	2211	508	1111	1575	0	591	575	0	0	0	0
0	0	0	0	0	5000	1743	2256	1080	1075	1681	0	561	501	0	591	0	0	0	0
0	0	0	0	5000	0	5000	5000	0	1582	567	1131	2227	5000	501	1667	5000	0	0	0
0	0	0	0	5000	5000	5000	5000	5000	0	0	567	1647	5000	0	5000	5000	0	0	0
0	0	0	5000	5000	D	5000	0	0	0	0	0	5000	5000	5000	5000	5000	5000	5000	0
0	0	0	5000	0	5000	5000	5000	0	0	0	0	0	0	5000	D	0	5000	0	0
0	0	0	0	5000	5000	5000	0	0	0	0	0	0	5000	5000	5000	5000	0	0	0
0	0	0	0	0	5000	0	0	0	0	0	0	0	0	5000	0	5000	0	0	0

Fig. 12. SECLOUD with multi-path and fake clouds

the benefits of the previous approaches, this approach further increases the unlinkability between the real source-destination pair.

D. Summary of Results and Comparison

Protocol	TO	Anonymity Level
SECLOUD single path	3.85	$1 - B^{-1}$
SECLOUD multipaths	3.82	$1 - B^{-1}$
SECLOUD single paths w. fake	7.69	$1 - [(\# of\, cloud\, pairs)B]^{-1}$
SECLOUD multipaths w. fake	7.53	$1 - [(\# of\, cloud\, pairs)B]^{-1}$

TABLE III
SECLOUD RESULTS SUMMARY

The overhead performance and anonymity level of several cases of SECLOUD from simulations are shown in Table III. 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.

Compared to fractal propagation where the minimum overhead is 475%, SECLOUD adds a fixed lower overhead as shown in Table III but provides better seclusion for both, source and destination nodes. The random walk protocol has an average overhead of 234% which is less that of SECLOUD. Combining random walk with cloud creation is a possibility under consideration.

In all the previous simulation results shown, we have used a grid network. However, for a random network in which the X and Y position of each node is independently and randomly using a uniform randomly distributed variable, the results of SECLOUD, Fractal and Random walk will all behave in a relatively similar manner. For brevity, we have not included the results in this paper.

Protecting privacy in the single source-destination case is more difficult compared to multiple pairs of source-destination. In the case of more than one real source-destination pairs, SECLOUD will have more pairs of clouds. If the attacker is targeting a specific source-destination, then the task of the attacker is also to select the correct source-destination from the correct cloud region. Note that to achieve privacy with more source-destination pairs, the sources have to start transmission roughly at the same time. Otherwise, a global attacker can monitor the network at time $T1$ when the first source-destination is activated and then monitor the network at time $T2$ when the second source-destination is activated.

VI. CONCLUSIONS

In this work, we presented a protocol to address the problem of source-destination privacy in the presence of a global and powerful attacker with network-wide traffic knowledge. We compare it with fractal propagation and random walk and show that it performs better than fractal propagation and favorably with random walk in terms of overhead, anonymity, and unlinkability.

REFERENCES

[1] A. A. Nezhad, A. Miri, and D. Makrakis, “Location privacy and anonymity preserving routing for wireless sensor networks,” *Comput. Netw.*, vol. 52, no. 18, pp. 3433–3452, 2008.

[2] J. Deng, R. Han, and S. Mishra, “Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks,” *Elsevier Journal of Pervasive and Mobile Computing on Security in Wireless Mobile Computing Systems*, vol. 2, no. 2, pp. 159–186, 2006.

[3] K. Mehta, D. Liu, and M. Wright, “Location privacy in sensor networks against a global eavesdropper,” in *Proc. of IEEE ICNP*, 2007.

[4] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, “Towards event source unobservability with minimum network traffic in sensor networks,” in *Proc. of ACM WiSec*, 2008.

[5] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, “A novel scheme for protecting receiver’s location privacy in wireless sensor networks,” *IEEE Trans. on Wireless Comm.*, vol. 7, no. 10, pp. 3769–3779, October 2008.

[6] Y. Xi, L. Schwiebert, and W. Shi, “Preserving source location privacy in monitoring based wireless sensor networks,” in *Proc. of SSN*, 2006.

[7] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, “Enhancing source-location privacy in sensor network routing,” in *Proc. IEEE ICDCS*, 2005.

[8] M. Shao, Y. Yang, S. Zhu, and G. Cao, “Towards statistically strong source anonymity for sensor networks,” in *Proc. of IEEE INFOCOM*, 2008.

[9] C. Ozturk, Y. Zhang, W. Trappe, and M. Ott, “Source-location privacy for networks of energy-constrained sensors,” in *Proc. of IEEE SEUS*, 2004.

[10] B. Hoh and M. Gruteser, “Protecting location privacy through path confusion,” in *Proc. of SECURECOMM*, 2005.

[11] M. Gruteser and D. Grunwald, “Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis,” *Mob. Netw. Appl.*, vol. 10, no. 3, pp. 315–325, 2005.

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

[13] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A key predistribution scheme for sensor networks using deployment knowledge,” *IEEE Trans. on Dependable and Secure Comp.*, vol. 3, no. 1, pp. 62–77, 2006.

[14] Y. Zhang, W. Liu, and W. Lou, “Anonymous communications in mobile ad hoc networks,” in *Proc. of IEEE INFOCOM*, 2005.

[15] J. Kong, X. Hong, and M. Gerla, “An identity-free and on-demand routing scheme against anonymity threats in mobile ad hoc networks,” *IEEE Trans. on Mob. Comp.*, vol. 6, no. 8, pp. 888–902, 2007.

[16] R. Shokri, M. Yabandeh, and N. Yazdani, “Anonymous routing in manet using random identifiers,” in *Proc. of the Sixth International Conference on Networking*, 2007.

[17] R. Lu, Z. Cao, L. Wang, and C. Sun, “A secure anonymous routing protocol with authenticated key exchange for ad hoc networks,” *Comput. Stand. Interfaces*, vol. 29, no. 5, pp. 521–527, 2007.

[18] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, “Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks,” in *Proc. of IEEE LCN*, 2004.

[19] D. Huang, “Traffic analysis based unlinkability measure for ieee 802.11b-based communication systems,” in *Proc. of ACM WiSe*, 2006.

[20] F. Kuhn and A. Zollinger, “Ad-hoc networks beyond unit disk graphs,” in *Proc. of the fifth international workshop on Foundations of mobile computing*, 2003.

[21] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–714, November 1986.