Is Random Walk Truly Memoryless - Traffic Analysis and Source Location Privacy under Random Walks

Rui Shi* Mayank Goswami† Jie Gao* Xianfeng Gu*

* Department of Computer Science, Stony Brook University {rshi, jgao, gu}@cs.sunysb.edu
† Department of Applied Mathematics, Stony Brook University {mayank.isi}@gmail.com

***Abstract*—Random walk on a graph is a Markov chain and thus is 'memoryless' as the next node to visit depends only on the current node and not on the sequence of events that preceded it. With these properties, random walk and its many variations have been used in network routing to 'randomize' the traffic pattern and hide the location of the data sources. In this paper we show a myth in common understanding of the memoryless property of a random walk applied for protecting source location privacy in a wireless sensor network. In particular, if one monitors only the network boundary and records the first boundary node hit by a random walk, this distribution can be related to the location of the source node. For the scenario of a single data source, a very simple algorithm which says the simple integration along the network boundary would reveal the location of the source. We also develop a generic algorithm to reconstruct the source locations for various sources that have simple descriptions (e.g., k source locations, sources on a line segment, sources in a disk). This represents a new type of traffic analysis attack for invading sensor data location privacy and essentially re-opens the problem for further examination.**

I. Introduction

Given a graph and a starting vertex, we choose a neighbor of the current node at random and move to this neighbor and continue in this fashion. This sequence of nodes is called a *random walk* on the graph. Random walk is a Markov chain such that the next node to visit only depends on the current node and is independent of the history. This is often termed as the "memoryless" property of a random walk, which makes it useful for many applications in computer networking. Of particular interest to this paper is the application of random walk in wireless sensor network routing for preserving source location privacy.

Source Location Privacy. Wireless sensor networks find many useful civilian and military applications. In many settings one would like to protect the privacy of sensor data, defined in the general sense that sensor data and its contextual information are observable by only those who are supposed to observe it [10]. Providing privacy in wireless sensor network is challenging for a number of reasons. Besides that the sensor nodes are low cost devices with limited computation and storage capacities, the fact that the sensor nodes use wireless medium make it susceptible to attacks such as eavesdropping and traffic analysis. In the literature, privacy threats in sensor networks are classified as content-oriented privacy threats (i.e., the leaking of packet content to adversaries), that can be addressed by security and encryption mechanisms, and contextual privacy issues (i.e., the leaking of context information related to the measurement and transmission of the sensor data), of which location of the data source is a major piece of information to be protected. In particular, an adversary may be able to compromise private information of source locations without the ability of decrypting the transmitted data – by simply monitoring and analyzing the traffic pattern in the air.

A classical model formed for protecting the source location privacy is the "Panda Hunter Game" [10]. In the game, a large number of panda detecting sensors are placed in a habitat to detect panda presence. Pandas here are analogs of generic assets to be monitored by a sensor network. When a panda is observed, the nearby sensor node will report such detection data periodically to the sink through multi-hop routing methods. The data package could be encrypted such that the adversary cannot decipher the content of the message and cannot derive the location of panda right away. However, an adversary, in this case, the hunter, can monitor the traffic in the network and by timing analysis trace back the routing path to the origin of the message, i.e., the location of the data source. Clearly, simple routing schemes such as shortest path routing cannot provide data source privacy against traffic analysis attacks.

Many schemes proposed in the literature for preserving source location privacy use a common idea of introducing randomness in packet routing. The objective is to make the traffic pattern look random and uncertain, and then counteract the adversarial traffic analysis attacks. Many of them use random walk or variations of random walks as a major component in the design. Phantom routing [10], for example, first uses random walk in the network until the node is reasonably far from the source node and then uses (probabilistic) flooding method to deliver it to the source. Although a short random walk may still have the current node correlated with the origin, a long random walk will stop at a location that is independent of the packet original. It is known that if the random walk is longer than the *mixing time*, the random walk converges to its limiting distribution called the stationary distribution [15]. This it is equivalent to selecting a node in the network randomly (from the stationary distribution) and thus packet analysis afterwards will only trace back to this random location, unrelated to the true data source.

Traffic Analysis on Random Walk. In this paper we show that it is a myth in common understanding that random walk automatically brings with it source location privacy. In other

words, we present a technique which allows certain traffic analysis to infer the source location even for random walks that are as long as they want. Therefore our message is that random walk should be used carefully in protecting source location privacy.

II. Overview

Network Model and Attack Model: We assume in this paper a wireless sensor network deployed in a planar domain $\mathcal{R}$ of interest for monitoring interesting events. The event locations are of great importance for both the network owners and the adversary. When an event is detected, the nearby sensor node becomes the data source and sends the report periodically to a data sink (e.g., a base station or a mobile sink) in the network. We assume that the message is delivered by using random walk, in which the next node to visit is uniformly chosen from all neighbors of the current node. The random walk is sufficiently long to ensure that the message will be delivered to the data sink with high probability. A data source will generate data packets periodically and the delivery of these packets is completely independent of each other. That is, they follow different random walk paths. The specific capabilities of the adversary is summarized below.

- *Monitoring traffic on network boundary.* We assume that the adversary can only monitor network traffic along the network outer boundary. This is a reasonable assumption in many settings when the domain of interest has restricted access to anyone but the network owner. It is also a realistic model of many military applications. The adversary places monitoring stations to monitor network traffic along the network outer boundary. Each monitoring station listens to the traffic in the neighborhood of a sensor node and record the signals delivered to/from the sensor node. We assume that the positions of the monitoring stations, or equivalently the network boundary, are known. The monitoring stations are also assumed to be perfectly synchronized. The traffic data from the monitoring stations is collected and delivered to an offline base station for further analysis. We remark that the assumption puts more restriction to the adversary's power than the Panda Hunter model, in which the adversary can be anywhere inside the network and can move around as fast as possible.
- *Packets are encrypted.* We assume that the packets in the network are encrypted using symmetric encryption between the data source and the data sink and that the adversary does not have the key to decipher the content of the message. Similar to the Panda Hunter problem, the data source issues data packets periodically. We assume that the content of these data messages are different, i.e., with different time stamps. The monitoring stations can compare the messages received by different boundary nodes and conclude whether two messages received by two boundary nodes are the same or not. We assume that the chained encryption scheme used in onion routing is not feasible for sensor network, for two reasons. First the chained encryption requires that the source knows the entire path taken by the message, which is not the case for random walk. Second, chained encryption and decryption for each relay node is too heavy for resource constrained sensor nodes.
- *Non-malicious.* The adversary does not interfere with the normal functioning of the sensor networks. Otherwise it will be detected by intrusion detection schemes. The adversary does not compromise any node and does not generate or alter traffic in the network.
- *Informed.* We use the standard philosophy in security [22] that the adversary is aware of the routing methods used by the system, in our case, the random walk scheme.
- *Centralized and powerful.* The monitoring stations gather traffic received from the network boundary and then deliver all the data to an offline central station for processing. We assume the adversary has abundant computing resources and can perform complicated analysis.

Traffic Analysis of Random Walk: We first consider a special case when the network is in a domain of disk shape and sensors are uniformly distributed inside the disk. In this case the random walk can be considered as a discrete approximation of the continuous Brownian motion inside a disk. For each message issued by the data source, through comparing the messages gathered by the monitoring stations at the network boundary we can conclude the node on the boundary that received the message for the first time. Now, since the data source generates multiple data packets, we monitor the position of the first hit on the boundary by different data packets. This constitutes a 'first hit' distribution (also called the exit distribution) ω'_x on the boundary where x is the source location. If the data source is at the center o of the disk, by symmetry the distribution ω'_x is a uniform distribution. When the data source is not at the center of the disk, the distribution has a single peak at the boundary intersected by the ray ox, and the closer the source to the boundary, the higher the peak is. See Figure 1 for an example. Therefore by monitoring the traffic pattern on the network boundary only, we obtain an observation of the first hit distribution p_x, through examining which we can infer the source location.

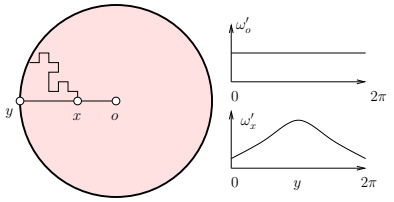

Fig. 1. The first hit distribution ω'_x and ω'_o for random walk inside a unit disk starting at x and o respectively.

In general the network may not be of a disk shape thus the first hit distribution could have a complicated correlation with the source location. For a bounded domain $\mathcal{R}$ in the plane, the probability that a Brownian motion started inside a point $z \in \mathcal{R}$ hits a portion of the boundary is termed the *harmonic measure* [9] ω_z. The first hit distribution observed from the traffic pattern ω'_z is a Monte Carlo approximation

of ω_z. On simply connected planar domains, there is a close connection between harmonic measure and the theory of conformal maps. A conformal map is a continuous one-to-one map that preserves angles. It is known that Brownian motions are conformally invariant [11]. What this means is that under a conformal map, $f : \mathcal{R} \to \mathcal{R}'$, the probability for a Brownian motion starting from $x \in \mathcal{R}$ and exiting from an interval $I[a, b]$ on the boundary $\partial\mathcal{R}$ is the same as the probability of a Brownian motion starting from $f(x) \in \mathcal{R}'$ and exiting from an interval $I[f(a), f(b)]$ on the boundary $\partial\mathcal{R}'$. See Figure 2 for an example. Now, since any simple planar domain can be mapped to a canonical shape of a unit disk by a conformal mapping, one can obtain the harmonic measure for any simply connected domain. In particular, take the example in Figure 1, we can apply a Mobius transformation f from a disk to a disk such that the point x is now mapped to the center of the disk. Therefore the distribution ω_x can be immediately computed through f.

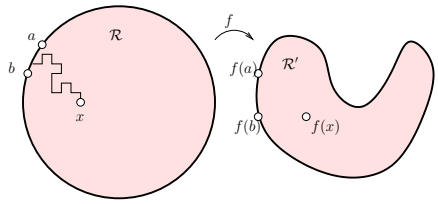

Fig. 2. The probability for a Brownian motion starting from $x \in \mathcal{R}$ and exiting from an interval $I[a, b]$ on the boundary $\partial\mathcal{R}$ is the same as the probability of a Brownian motion starting from $f(x) \in \mathcal{R}'$ and exiting from an interval $I[f(a), f(b)]$ on the boundary $\partial\mathcal{R}'$.

The discussion above suggests that the exit distribution observed by the adversary along the sensor network boundary can be used to infer the source locations. In this paper we present such traffic analysis algorithms. We present two algorithms specifically. The first one is for recovery of a single data source. It is very simple, by integrating the position and the harmonic measure along the domain boundary, i.e., $\int_{z\in\partial\mathcal{R}} z d\omega_x(z)$. To understand this, take a look at Figure 1. If the source is at o and we integrate the position by the harmonic measure ω_o (which is uniform) along the unit circle, by symmetry this integration gives us the center of the disk. If the source is at x, the integration of the position by ω_x must lie on the line segment oy – again by axial symmetry of ω_x with respect to oy. In fact, this integration would give precisely the position of x. And this is true not only for the case of a unit disk but for *any* planar domain. Since the first hit distribution observed from the traffic pattern, ω'_x, would be a good approximation to the harmonic measure ω_x. By using $\int_{z\in\partial\mathcal{R}} z d\omega'_x(z)$ we will get a very close approximation to x, as long as we have enough samples to be statistically meaningful.

The second algorithm is a general method using maximum likelihood estimation and it can be used for a general case when the data sources can be represented using low complexity. A number of representative scenarios include multiple data sources, data sources uniformly distributed on a line segment, as in the case of target tracking applications, or data sources uniformly inside a small disk or square, as in the case when an event triggers multiple sensors to report to the sink. The results and the algorithms can be extended to a non-simple planar domain as well as a general non-planar terrain.

We presented an extensive list of simulations for different network shape and different data source models as mentioned above. In particular, we presented the tradeoff between the number of messages issued by the data source vs the accuracy of our prediction of the source location.

Last we want to remark that we do not mean to claim that previous source location privacy preserving schemes using random walks are inadequate, but rather raise an alarm that their effectiveness should be reconsidered carefully given the potential attack illustrated in this paper. At the end of the paper we discuss variations of basic random walks and suggest ideas to defeat this particular traffic analysis attack.

III. Theory

In this section we first summarize the main results from the elegant theory of Brownian motions and conformal maps. We then provide the background on random walks in the discrete setting, and state our results.

Conformal Maps:

Let $\mathbb{C} = \{z : z = x + iy;\ x, y \in \mathbb{R}\}$ denote the complex plane. The following material can be found in [1], [6].

Definition 3.1. *A holomorphic function f on a domain $D \subset \mathbb{C}$ is a complex valued function defined on D such that the complex derivative of f exists everywhere inside D. This also implies that f is infinitely differentiable, equal to its own Taylor series and preserves angles at all points where the derivative of f is non-zero.*

A holomorphic function which has a non-zero derivative everywhere is also called conformal.

Definition 3.2. *A harmonic function f on a domain $D \subset \mathbb{R}^2$ is a twice continuously differentiable real valued function such that $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$.*

Here are two useful properties:

- Let $f(z) = f_1(z) + if_2(z)$ be holomorphic. Then f_1 and f_2 are harmonic.
- *Mean Value Property* Let u be holomorphic/harmonic on the unit disk $\mathbb{D}$. Then, $u(0) = \int_{\partial\mathbb{D}} u(e^{i\theta}) \frac{d\theta}{2\pi}$.

Mobius transforms and Riemann mapping:

Let $\mathbb{D}$ denote the unit disk centered at the origin in $\mathbb{C}$. The group of Möbius transformations is the set of all conformal maps from $\mathbb{D}$ to itself. It is well-known that any such map is of the form $f(z) = e^{i\theta} \frac{z - z_0}{1 - \bar{z}_0 z}$ for some $\theta \in (0, 2\pi)$ and some $z_0 \in \mathbb{D}$.

Let Ω be a simply connected domain (a topological disk) in the plane, such that the boundary $\partial\Omega$ is a smooth curve:

Theorem 3.3 (Riemann Mapping). *Let Ω be as above. Then there exists a conformal map $f : \mathbb{D} \longrightarrow \Omega$. Further, f is unique upto composition by a Möbius transformation.*

Harmonic Measure:

Definition 3.4 (Harmonic Measure). *[2] [7] For any subset X of the boundary ($X \subset \partial\Omega$), the harmonic measure of X with respect to z is defined as $\omega(X, \Omega, z) = \frac{1}{2\pi}|f^{-1}(X)|$.*

Here $|.|$ denotes the Euclidean length of an arc on the unit circle. Note that any two conformal maps sending O to z only differ by a rotation, so this definition does not depend on the f chosen. Using harmonic measure, one can extend the Mean-value property to arbitrary domains. If u is a harmonic function on an arbitrary simply connected domain Ω, $z_0 \in \Omega$ is a base point and f_{z_0} is a conformal map such that $f(0) = z_0$, then $u \circ f$ is harmonic on the disk, so that

$$u(z_0) = (u \circ f)(0) = \int_{S^1} u(f(e^{i\theta}))\frac{d\theta}{2\pi} = \int_{\partial\Omega} u(z)d\omega_{z_0} \quad (1)$$

where $d\omega_{z_0}$ is the harmonic measure with respect to z_0.

The harmonic measure $\omega(X, \Omega, z)$ is related to a Brownian Motion started in the domain Ω frm the point z. We define Brownian Motion next.

Brownian Motion:

Definition 3.5. *A one-dimensional Brownian Motion [12] W_t intuitively is a scaling limit of the random walk. In other words, it is a stochastic process indexed by time $t > 0$, which has the following properties :*

1) *$W_0 = x$; here $x \in \mathbb{R}$ is the starting point.*
2) *The process has independent increments, i.e. for any two disjoint intervals $[s_1, t_1]$ and $[s_2, t_2]$, where $s_i, t_i > 0$, the increment in one interval $W_{t_1} - W_{s_1}$ is independent of the increment in the other $W_{t_2} - W_{s_2}$.*
3) *$W_{t+h} - W_t$ is Normally distributed with mean 0 and variance h.*
4) *Almost surely, the function $t \longrightarrow W_t$ is continuous.*

The case $W_0 = 0$ is called Standard Brownian Motion. A two-dimensional Brownian motion is a pair $B_t = (W_t^1, W_t^2)$ of two independent one-dimensional Brownian Motions.

Harmonic Measure, Brownian Motion and Conformal Invariance:

An important property of the Brownian motion is that it is invariant under conformal changes, i.e. the image of a Brownian motion under a conformal map is again a Brownian motion in the image of the domain [12]. The Brownian Motion can be viewed as the limit, as $t \longrightarrow 0$, of a walk which starts at 0, chooses a direction randomly, goes a distance t in that direction, and continues this way at every point. The angle changes are preserved under conformal maps, therefore one should expect that the law of the trajectory should be invariant.

Clearly, the same is true for harmonic measure. In other words, $\omega(X, \Omega, z) = \omega(f(X), f(\Omega), f(z))$ for any $X \subset \partial\Omega$ and f conformal.

Discrete Theory:

In this section, we summarize the related theories of random walks on graphs.

Suppose G is a planar graph, embedded on the plane. Let $V = \{v_1, v_2, \cdots, v_n\}$ be the vertex set, (x_k, y_k) be the 2D position of vertex v_k, $E = \{e_1, e_2, \cdots, e_m\}$ be the edge set. For simplicity, we assume each face of G is a triangle. The following edge weight definition is motivated by the

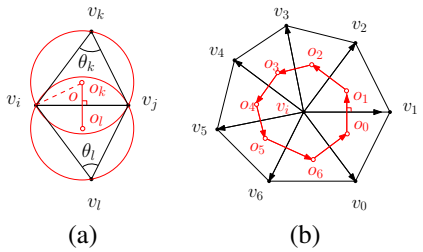

Fig. 3. (a) shows the edge weight. (b) shows that the vertex position function is harmonic.

relationship of random walk and resistance of the triangulation as in an electrical network [3] [5].

Definition 3.6 (Cotangent Edge Weight). *[3], [5] Suppose edge $[v_i, v_j]$ is adjacent to two faces $[v_i, v_j, v_k]$ and $[v_j, v_i, v_l]$, then the weight on edge is given by $w_{ij} = \frac{1}{2}(\cot\theta_k + \cot\theta_l)$.*

The edge weight determines the transition probability for a random walk on graph.

Definition 3.7 (Random Walk on Graph). *Suppose $X(t)$ is a random walk on the graph G defined as follows: if at time t the walk is at vertex v_i, then the probability of v_j being the next vertex is given by: $Prob\{X(t+1) = v_j | X(t) = v_i\} = \frac{w_{ij}}{\sum_k w_{ik}}$.*

When we choose a uniform sampling and all the triangles are equilateral triangles, all the edge weights are close to 1. In this case the above definition becomes the same as the random walk with uniform distribution on all neighbors. In our simulations we choose G to be a Delaunay triangulation on a nice set of samples inside $\mathcal{R}$.

Definition 3.8 (Discrete Harmonic Measure). *Suppose G is a planar graph with triangular faces. If the random walk $X(t)$ starts from a vertex v_i and exits at $v_k \in \partial G$, then the discrete harmonic measure is defined as the probability $\omega_k(v_i) := Prob\{X \sim v_k | X(0) = v_i\}$.*

Here $X \sim p$ means that the random walk X exits the boundary ∂G via the point p.

Definition 3.9 (Discrete Laplace Operator). *Let $f : V \to \mathbb{R}$ be a function defined on the vertices of the graph G. The discrete Laplace operator is defined as $\Delta f(v_i) = \sum_j w_{ij}(f(v_j) - f(v_i))$.*

Definition 3.10 (Discrete Harmonic Function). *Let $f : V \to \mathbb{R}$ be a function and Δ be the discrete Laplace operator. If Δf equals to zero for all vertices, then f is called a discrete harmonic function.*

From definition, it is easy to show that discrete harmonic measures $\omega_j : V \to \mathbb{R}$, $\forall v_j \in \partial G$ are harmonic functions. By definition, expected position function is harmonic. Figure 3 shows the vertex position function is also harmonic. Like smooth case, discrete harmonic functions have mean-value property, which states the value at each vertex is the average of

the values in the neighborhood. Mean-value property implies maximal value principle, which says the max and min value of a harmonic function must be on the boundary of the graph.

Definition 3.11 (Discrete Dirichlet Problem). *Suppose $f : V \to \mathbb{R}$ is a function defined on the graph, f is harmonic, and with boundary condition $f|_{\partial G} = g$,*

$$\begin{cases} \Delta f(v_i) &= 0 & \forall v_i \notin \partial G \\ f(v_j) &= g(v_j) & \forall v_j \in \partial G. \end{cases} \quad (2)$$

Then from the maximum modulus principle, we can get the uniqueness of the solution to the discrete Dirichlet problem. The solution to the Dirichlet problem can be explicitly given using harmonic measure.

Theorem 3.12 (Harmonic Measure Boundary Integration). *Suppose $f : V \to \mathbb{R}$ is the solution to the Dirichlet problem (Eqn.(2)). Then $f(v_i) = \sum_{v_j \in \partial G} g(v_j)\omega_j(v_i)$.*

Suppose a vertex v_0 at (x_0, y_0) sends messages routed by random walks. Figure 3 (b) shows the position function is harmonic. According to theorem 3.12, $(x_0, y_0) = \sum_{v_k \in \partial G}(x_k, y_k)\omega_k(v_0)$.. This is a linear running time algorithm, given the harmonic measure $\omega_k(v_0) = \text{Prob}\{X \sim v_k | X(0) = v_0\}$. In our applications, we estimate the harmonic measure simply by the ratio between the number of messages received at v_k and the total number of messages.

The above definitions and theorems do not require the graph to be planar. In fact, these concepts can be defined on triangular meshes in $\mathbb{R}^3$. But the 3D vertex position is not harmonic. Similar to smooth case, one can apply conformal mapping [19] [8] to flatten the 3D triangulation and use the same method to estimate the source position on the 2D image. Because the Laplace matrix is solely determined by the connectivity of the graph and the corner angles, roughly speaking, discrete conformal mapping preserve angles, therefore conformal mapping preserves harmonic measures. Therefore, the harmonic measure can be estimated using the random walks on the 3D mesh, and applied for boundary integration to estimate the source location on the 2D image plane.

IV. Traffic Analysis on Random Walks

A. Settings

We assume that a sensor network W is deployed densely in a geometric domain $\mathcal{R}$. Packet routing in the sensor network is done by random walk on the network. Suppose that a data source at x generated N data messages, we record for each message the boundary node that receives this message for the first time. This frequency count can be normalized as a distribution ω'_x on the sensor network boundary. The input to the traffic analysis algorithm for the adversary is the exit distribution ω'_x, together with the geometry of the sensor network boundary $\mathcal{R}$. The adversary has no knowledge of the sensor network in the interior of $\mathcal{R}$ and would like to reconstruct the position x.

To reconstruct the source location, we assume that the sensor network is dense and thus the random walk is a good approximation of Brownian motion in the continuous domain $\mathcal{R}$. Therefore, for each point $x \in \mathcal{R}$, define by ω_x the exit distribution of Brownian motion starting from x. We will compare ω'_x to ω_x to reconstruct the position of the source. Notice that in this setting there are two relaxations: 1) the distribution ω'_x is obtained through random walk on the (unknown) graph W; 2) the distribution ω'_x is obtained through a Monte Carlo method, i.e., based on the frequency count of N random walk samples. Thus our prediction of the source location could be a bit off from the true source location. But if random walks on the real sensor network are good approximations of the Brownian motion in $\mathcal{R}$, and that the number of samples, N, is not too small, the error in the prediction is expected to be small. This is indeed confirmed by simulations in the next section.

We will present two algorithms. The first algorithm provided a closed-form solution by simply integrating along the domain boundary $\mathcal{R}$. It works for a single source on a topological disk domain or topological disk with multiple holes. The second algorithm is based on maximum likelihood method. Basically by comparing ω' and ω (the exit distribution of brownian motion), we find the source location y such that ω'_x and ω_y are the most similar. This is a generic framework for finding the locations of multiple data sources or any sources that can be represented in a compact way.

B. ALG1: Integration Along Domain Boundary

Recall that if u is a harmonic function on the domain Ω, then its value at any point in the interior can be recovered by its values on the boundary, as long as one knows the harmonic measure of the boundary, i.e. $u(z_0) = \int_{\partial\Omega} u(z) d\omega_{z_0}$ where $d\omega_{z_0}$ is the harmonic measure with respect to z_0. Clearly, the identity function $u(z) = z$ is holomorphic (i.e., is differentiable in z), the real part and imaginary part are both harmonic. Hence we get $z_0 = \int_{\partial\Omega} z d\omega_{z_0}$.

For the case of a single source at position z, our construction algorithm is to simply multiply the coordinates of the location of a point $p \in \partial\mathcal{R}$ with its harmonic measure and add the resultants over the entire boundary. This algorithm is a linear running time algorithm with complexity dependent only on the *length* of the boundary $\partial\mathcal{R}$. The algorithm applies for all planar domains, including multiply connected ones.

Calculating harmonic measure Now we show how to efficiently compute $\omega(X, \mathcal{R}, z)$,i.e. for any point z and any subset X of the boundary of $\mathcal{R}$, the probability that a random walk started from z will first exit the boundary from X. We first handle the (highly symmetric) case where the domain is the disk $\mathbb{D}$; X then is a subset of the unit circle and the starting point is the origin.

$\omega(X, \mathbb{D}, 0)$: This is the probability that a random walk started from the origin in the disk exits the disk from the set X on the boundary. Clearly, this is uniform (by symmetry), and hence $\omega(X, \mathbb{D}, 0) = \frac{|X|}{2\omega}$. In other words this probability is just the normalized Euclidean arclength of X.

$\omega(X, \mathbb{D}, z_0)$: To compute the harmonic measure for an arbitrary point $z_0 \in \mathbb{D}$, recall from III that the (conformal)

Möbius transformation $g(z) = \frac{z-z_0}{1-\bar{z}_0 z}$ maps the unit disk to itself and sends the point z_0 to the origin. Now, we use the property that the harmonic measure is preserved under conformal maps to obtain

$$\omega(X, \mathbb{D}, z_0) = \omega(g(X), \mathbb{D}, g(z_0)) = \omega(g(X), \mathbb{D}, 0) = \frac{|g(X)|}{2\omega}$$

$\omega(X, \mathcal{R}, z_0)$ **for arbitrary** $\mathcal{R}$ Here we will describe how to find the harmonic measure for an arbitrary planar domain $\mathcal{R}$. The first method only works for simply connected domains (domains with no holes) while the second works for both simply and multiply connected domains.

Method 1: Using Riemann Mapping This method uses the conformal invariance we described in Section III. As above, let $\mathcal{R}$ be a simply connected domain, with boundary Γ a Jordan curve. In almost all practical applications, one approximates $\mathcal{R}$ by a polygon, and Γ by a polygonal chain. The first step is to compute the Riemann mapping from D to $\mathcal{R}$. For accomplishing this task, various methods have been proposed [19] [8].

So let us assume we have computed the Riemann mapping $f : \mathbb{D} \longrightarrow \mathcal{R}$. Notice that $f^{-1} : \mathcal{R} \longrightarrow \mathbb{D}$ is also conformal and once again, conformal invariance implies that $\omega(X, \mathcal{R}, z_0) = \omega(f^{-1}(X), \mathbb{D}, f^{-1}(z_0))$ and we have shown how to compute $\omega(X, \mathbb{D}, z)$ for arbitrary $X \subset \partial\mathbb{D}$ and $z \in \mathbb{D}$ previously.

Method 2: Symm's Method This method does not require one to explicitly compute the Riemann Mapping from $\mathbb{D}$ to $\mathcal{R}$, and holds for multi-holed domain. We refer the reader to [2] for a short summary of this method.

Recall from 1 that for any holomorphic function u on $\mathcal{R}$, we have the property $u(z_0) = \int_{\partial\mathcal{R}} u(z) d\omega_{z_0}$. We can discretize the boundary of $\mathcal{R}$ into n intervals $\{P_j\}_{j=1}^n$, assume that the harmonic measure is constant in each interval and look at the discrete counterpart to the above equation:

$$u(z_0) = \sum_j \int_{P_j} u(z) d\omega_{z_0} = \sum_j \frac{\omega_{z_0}(P_j)}{|P_j|} \int_{P_j} u(z) dz$$

Now if we choose n independent harmonic functions $\{u_i\}_{i=1}^n$, we get a system of n equations in n unknowns and we can solve to find $\omega_{z_0}(P_j)$.

C. ALG2: Maximum Likelihood Method

To apply a maximum likelihood approach (MLE), we first need the exit distribution/harmonic measure of a Brownian motion starting at a point $z \in \mathcal{R}$, which can be computed using methods in the section above. We then explain the application of MLE for different settings.

Let $f(.|\theta)$ denote a family of distributions parameterized by θ. If one observes an i.i.d. sample $x_1, x_2, ...x_n$ from one of the distributions in this family, the Maximum Likelihood Method is a way to estimate the true parameter θ_0 such that this sample is most likely to come from $f(.|\theta_0)$.

Since the observations are assumed to be identically and independently distributed, the joint density function is

$$f(x_1, x_2, ...x_n|\theta) = f(x_1|\theta) f(x_2|\theta) ... f(x_n|\theta)$$

One then forms the *Likelihood Function*

$$\ell(\theta|x_1, x_2, ...x_n) = \Pi_{i=1}^n f(x_i|\theta)$$

The maximum likelihood estimate (MLE) $\hat{\theta}$ is defined to be the value of θ which maximizes the likelihood function, given the observed values x_i, i.e.

$$\hat{\theta} = \arg\max_\theta \ell(\theta|x_1, x_2, ...x_n)$$

For simplicity, the log-likelihood function $\hat{\ell} = \log \ell$ is also used, since log is a monotonic transformation.

From now on, $f_z := f(x|z)$ will denote the density function for the harmonic measure. Denote by X_z the exit position (the first hit position) of a random walk starting at z. It is a random variable distributed with density f_z; $\mathbb{P}(X_z \in A) = \int_A f_z(x) dx$ for all $A \subset \partial\Omega$.

- **Single source.** Suppose that $x_1 x_2, ...x_N$ are the first hit positions on the boundary for the N messages sent by an unknown source $z_0 \in \mathcal{R}$ respectively. We know $f(x|z)$ from the previous section, form the likelihood function and maximize.
- k **sources,** k **is known.** This boils down to the single source problem for each of the sources. Now let's assume that the adversary cannot distinguish the data packets from different sources. Let the unknown source locations be $z_1, ...z_k$. Then what we observe is the random variable

  $$Y = X_{z_1} + X_{z_2} + ...X_{z_k}$$

  Given the z_i, the density of Y can be computed. Again one can form the likelihood function and maximize, now with respect to the vector of z_i. We also allow short-lived fake message which is sent to a randomly selected neighbor by the relay node after a real message is relayed. Our traffic analysis is not affected if the fake messages are discarded and not relayed any further.
- **Source moving on a line.** Assuming that we have a mobile data source moving on a line. The source sends packets periodically after distance ϵ. We are interested in estimating the initial position z_0 and the direction θ in which the source is travelling. Let $z_i = z_0 + i\epsilon e^{i\theta}$. Notice here we just need to estimate 3 real parameters, thus we could expect to get good estimates with just a lot fewer data packets per source z_i.

V. Simulations

We conducted extensive simulation tests to examine the performance of our algorithm to find the source location, as well as how recovery accuracy is affected by different parameters.

The simulations were done under different settings, namely a unit disk, a planar non-disk domain, a planar domain with holes and a non-planar domain. Also for each type of domain, we conducted simulations using both a triangle mesh (TM) and a unit disk graph (UDG). In TM model, we calculated the transition probability for each node d by it's neighbors in the triangulations; for UDG model, we calculated the transition

probability for d by it's neighbors in the unit disk graph. We scaled all planar domains inside a 2×2 bounding box, and scaled non planar domains inside a $2 \times 2 \times 2$ bounding box. We use the term $Error$ to measure the distance between the true source location and the location predicted by our algorithm. The $Error_{ave}$ and $Error_{max}$ bellow, which represent the average and max value of $Error$, are respect to the bounding box unit above. In the following, N_{domain} represents the number of nodes inside domain R, N_{msg} represents the number of messages issued at each source node.

Unit Disk Domain Figure 4 right and figure 5 right show the relationship between N_{msg} with $Error_{ave}$ and $Error_{max}$ under TM disk model and UDG disk model respectively. This is obtained by fix N_{domain}=1K, then randomly chose n=100 sources inside the R and issued N_{msg} numbers of random walks started from each of these chosen sources, then calculated the $Error_{ave}$ and $Error_{max}$ respectively. Beside this, we also examined how the location of source (the distance r from disk center) affects $Error_{ave}$. We uniformly sampled $0 < r < 1$ to get $\{r_1, r_2, ...r_m\}$, for each r_i we randomly chose n_i=100 points whose distance to center r_{n_i} satisfies $r_i - \varepsilon < r_{n_i} < r_i - \varepsilon$ (here we used ε=0.05) as the source to issue random walk for N_{msg}=1000 times. Then we use our method to predict the source location according to the boundary message distribution. Based on the real source location and the one calculated by our method, we computed $Error_{ave}$ for each r_i. Figure 4 left and figure 5 left show the relationship between r_i and $Error_{ave}$ under TM model and UDG model respectively. We can see that $Error_{ave}$ decreased while the real source leaving the disk center.

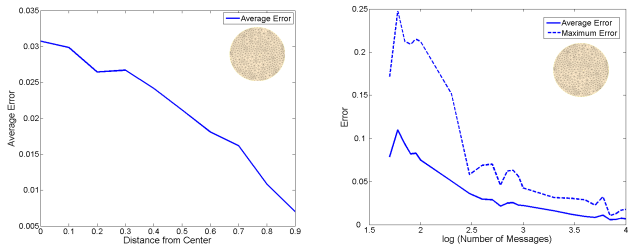

Fig. 4. Left: Distance from Center VS. $Error_{ave}$ under TM Model. Right: N_{msg} VS. $Error_{ave}/Error_{max}$ under TM Model.

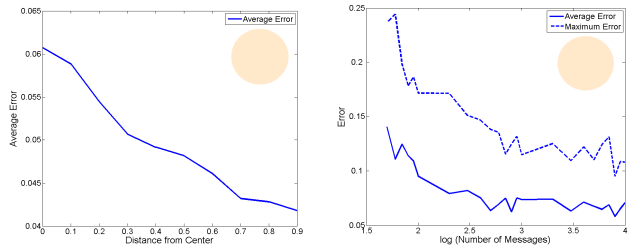

Fig. 5. Distance from Center VS. $Error_{ave}$ under UDG Model. Right: N_{msg} VS. $Error_{ave}/Error_{max}$ under UDG Model.

Planar non-disk Domains We did the same kind of simulation on an irregular domain. We evaluated how N_{msg} affects $Error_{ave}$ and $Error_{max}$ by fix N_{domain}=1K. The results are shown in figure 6. We can see that $Error_{ave}$ and $Error_{max}$ decreased while we increased N_{msg}. We obtained $Error_{ave}$ around 0.04 and 0.08 under TM model and UDG model by 100 messages.

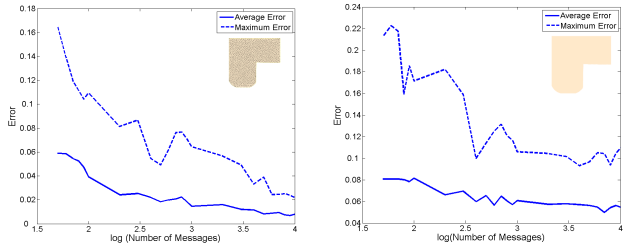

Fig. 6. Left: N_{msg} VS. $Error_{ave}/Error_{max}$ under TM Model. Right: N_{msg} VS. $Error_{ave}/Error_{max}$ under UDG Model.

Planar Domain with Holes The same as above, we evaluated how N_{msg} affects $Error_{ave}$ and $Error_{max}$ for a planar domain with holes. For a planar domain with holes, as long as we can monitor the inside hole boundaries as well, we can just treat them as the same as outer boundary in the calculation. The results are shown in figure 7. We obtained $Error_{ave}$ around 0.04 and 0.07 under TM model and UDG model by 100 messages.

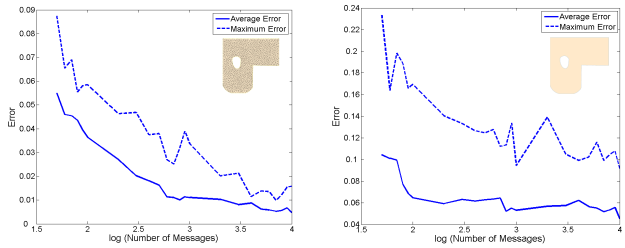

Fig. 7. Left: N_{msg} VS. $Error_{ave}/Error_{max}$ under TM Model. Right: under UDG Model.

Non-planar Domain For a general non-planar domain, we first mapped it to the unit disk using conformal mapping method in [8]. Since Brownian motion is invariant under conformal mapping, we used the same method to calculate source location in the parameter domain, then mapped it back to the original surface. The simulation results are in figure 8. We obtained $Error_{ave}$ around 0.08 and 0.09 under TM model and UDG model by 100 messages.

Visualization of Exit Distribution Following we show the exit distribution along the domain boundary. We took the non-uniform planar domain, set an arbitrary source and visualizes the exit distribution (figure 9 left) using small disks along the boundary with area proportional to $NO.\ of\ first\ hit.$ We also show the distribution on the parameter domain, which is obtained by conformally mapping the non-uniform domain to a unit disk (figure 9 right). The distribution on the parameter domain gives strong evidence that conformal mapping preserves Brownian motion. Namely the Brownian motion starting from source s on surface M is equivalent

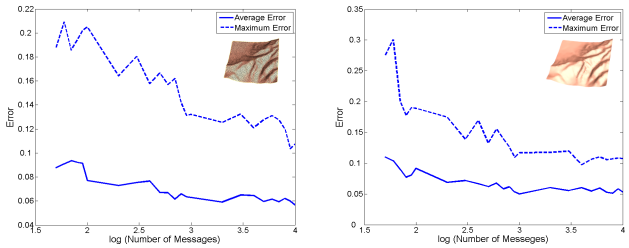

Fig. 8. Left: N_{msg} VS. $Error_{ave}/Error_{max}$ under TM Model. Right: under UDG Model.

to the Brownian motion start from $\phi(s)$ on surface $\bar{M}$, if $\phi : M \to \bar{M}$ is a conformal mapping from M to $\bar{M}$.

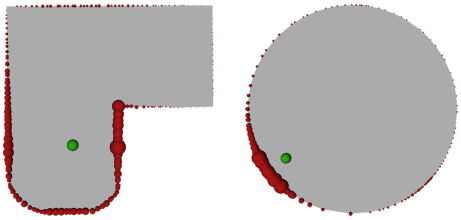

Fig. 9. Left: First Hit Distribution. Right: First Hit Distribution on parameter domain.

Network Density Versus Average Error To examine how much the network density N_{domain} affects the average distance error $Error_{ave}$ by fix N_{msg}, then varying N_{domain} and observe $Error_{ave}$. The results are shown in Figure 10.

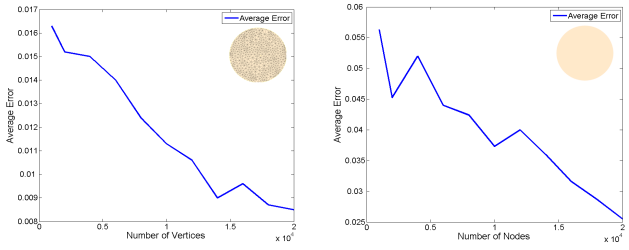

Fig. 10. Left: N_{domain} VS. $Error_{ave}$ under TM. Right: N_{domain} VS. $Error_{ave}$ under UDG.

Multiple Sources We uniformly discretized the unit square domain into $N \times N$ grids(N=20 in our experiment), and assumed the possible location of a source is on the center of a grid. For 2 sources case, there are $N^4/2$ numbers of possible source location combinations. For each possible pair $(s_i,\ s_j)$, we issued $N_{msg} = 2000$ numbers of random walks from s_1 and s_1 , then stored a set of first hit distributions $\{\Phi_{ij},\ 0 < i, j < N\}$. Then we randomly picked sources pair $(s_1,\ s_2)$ to issue $\bar{N}_{msg}$ random walks and obtained a first hit distribution Φ_{test}. By comparing Φ_{test} with Φ_{ij} we got a p-value which stands for the probability that Φ_{test} and Φ_{ij} are the same distribution. The i, j which gave the maximum p-value directly points out the location of s_i and s_j. In this experiment, we varied $\bar{N}_{msg}$ and obtained a set of $Error_{ave}$, like in figure 11. We can see that $Error_{ave}$ decreased as we increased $\bar{N}_{msg}$.

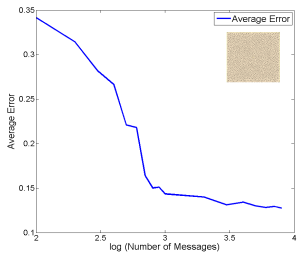

Fig. 11. N_{msg} VS. $Error_{ave}$ for two sources.

VI. Related Work

Routing that preserves source anonymity has been a topic of study for a number of years. For routing on the Internet, one would like to hide the sender's identity, as phrased in anonymous routing. The most popular schemes are Chaum's mixes [4] and onion routing [20], [21]. In Chaum's scheme, the idea is to send the message in an encrypted manner to a central server called the anonymizer, which removes the source identity and then sends the message to the receiver. Thus one cannot differentiate the sources of the messages delivered by anonymizer. Onion routing uses encryption on source routing, such that the source identifies the entire routing path to the destination and encrypt the messages in layers in the order of the nodes along the path. Each relay node descrypt the message using its own private key, which reveals the next hop and sends the message. In this way each node on the path is only aware of the immediate upstream and downstream node and is not aware of the entire path, in particular the source identity. Both schemes cannot be applied in sensor network setting since we cannot afford a central server, and public key encryption is too heavy for sensor nodes. In addition, encryption based security schemes only protect the content of the messages but cannot deal with traffic analysis attacks.

Existing schemes for preserving source location privacy is summarized in a recent survey [13]. Among them, random walk is a commonly used component. Phantom routing [10], [18] first uses random walk to arrive at a node that is reasonably far away from the source and then use probabilistic flooding to deliver the message to the destination. Followup schemes such as in [14], [16], [23] use biased random walk in order to get farther away from the data source, or introduce fake data sources to further confuse the traffic pattern [10], [17]. In the next section we examine some of these variations and discuss the performance of the traffic analysis attack for these cases.

VII. Discussions

Length of Random walks Our traffic analysis scheme uses the exit distribution of random walks on the network boundary. This means that the random walks should be long enough so that they hit the network boundary with good probability before they stop. We argue that this is true as the random walks should be long enough to deliver the message to the data sink. If the data sink is at an unknown location in the

network, the random walk should be long enough so that it visits every node in the network. This is termed as the *cover time*, defined as the expected number of steps for a random walk to cover all the nodes in the network [15]. For a 2D grid of n nodes the cover time is roughly in the order of $\Omega(n^2)$.

To estimate the probability that a random walk of length h hits the network boundary, we again consider a 2D grid of n nodes. Suppose X_i is the displacement vector of the i-th step of the random walk. X_i is uniformly chosen from $\{(1,0),(-1,0),(0,1),(0,-1)\}$. The position of random walk after i steps starting from the center of the grid is simply $P_i = X_1 + X_2 + \cdots + X_i$. By the central limit theorem, P_i is a Gaussian distribution with mean $(0,0)$ and variance $h/2\mathbb{I}$, where I is a 2×2 identity matrix. Thus the probability that P_i is outside a disk of radius r from the center is estimated as $e^{-r^2/h}$. Choose h to be $O(n^2)$ and r to be $\sqrt{n}$, the probability above is $1 - 1/n$. This means that the random walk of length $O(n^2)$ has a high probability to hit the network boundary at least once. This means that for a random walk to deliver the message to the sink, it must hit the boundary with high probability. This assures that the traffic analysis along the boundary could be performed.

Directed or Biased random walk In a standard random walk, the next node to visit is chosen *uniformly randomly* from all neighbors. This is the discrete analog of Brownian motion which is isotropic. The first variation to it is to define a non-uniform probability distribution on neighbors. In Phantom routing and a number of followup papers, a biased random walk is often adopted in which the neighbor that is farther away from the data source is chosen with higher probability, in order to quickly get to the regions far away from the data source. For example, in sector-based directed random walk [10], a random walk from the west will be sent to a node to the east, chosen uniformly randomly. In hop-based directed random walk [10], [16], a random walk chooses the next hop uniformly randomly among only the nodes closer to the sink.

If the transition probability is non-uniform but determined (as in the two cases mentioned above), the harmonic measure as defined by the random walk will change. If the transition probability is known to the adversary, we can still calculate the harmonic measure under this change. Using the same idea presented in the paper one can still infer the source location. Therefore to make a biased random walk to be a countermeasure of the traffic analysis, we need to make the transition probability to be *unknown* to the adversary. One idea is to vary this transition probability randomly and periodically. However, in this case one should be careful about the transition probability configuration to make sure that the random walk is still ergodic[1] – otherwise there is no guarantee that the random walk covers the entire network and eventually delivers the message to the data sink.

[1]A random walk is ergodic when there is a unique stationary distribution. This requires the graph (implied by the edges with non-zero transitional probability) to be connected and non-bipartite.

VIII. Conclusion

In this paper we show a traffic analysis scheme such that an adversary can infer the location of the data source issuing packets routed by random walks in a sensor network. Since random walk has been used as a common component in most of previous work in preserving source location privacy, this re-opens the question as how to best protect the source location privacy. We consider this as our future work.

References

[1] L. Ahlfors. *Lectures in Quasiconformal Mappings*. Van Nostrand Reinhold, New York, 1966.

[2] C. Bishop. The riemann mapping theorem. http://www.math.sunysb.edu/ bishop/classes/math401.F09/t.pdf.

[3] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 2002.

[4] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.

[5] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. The Mathematical Association of America, 1984.

[6] H. M. Farkas and I. Kra. *Riemann Surfaces*. Springer, 2004.

[7] J. Garnett and D. Marshall. *Harmonc Measure*. Cambridge University Press, 2005.

[8] X. Gu and S.-T. Yau. Global conformal parameterization. In L. Kobbelt, P. Schröder, and H. Hoppe, editors, *Symposium on Geometry Processing*, volume 43 of *ACM International Conference Proceeding Series*, pages 127–137. Eurographics Association, 2003.

[9] S. Kakutani. On brownian motion in n-space. *Proc. Imp. Acad. Tokyo*, 20(9):648–652, 1944.

[10] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, ICDCS '05, pages 599–608, 2005.

[11] G. Lawler. *Conformally Invariant Processes in the Plane*. Amer Mathematical Society, 2005.

[12] G. F. Lawler. Conformally invariant processes in the plane. *Mathematical Surveys and Monographs*, 114(2), 2008.

[13] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham. Privacy preservation in wireless sensor networks: A state-of-the-art survey. *Ad Hoc Netw.*, 7:1501–1514, November 2009.

[14] Y. Li and J. Ren. Preserving source-location privacy in wireless sensor networks. In *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON'09, pages 493–501, 2009.

[15] L. Lovasz. Random walks on graphs: A survey. *Bolyai Soc. Math. Stud.*, 2:353–397, 1996.

[16] X. Luo, X. Ji, and M.-S. Park. Location privacy against traffic analysis attacks in wireless sensor networks. In *2010 International Conference on Information Science and Applications*, pages 1–6. Ieee, February 2010.

[17] K. Mehta, D. Liu, and M. Wright. Protecting location privacy in sensor networks against a global eavesdropper. *IEEE Trans. Mob. Comput.*, 11(2):320–336, 2012.

[18] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, SASN '04, pages 88–93, 2004.

[19] R. Sarkar, X. Yin, J. Gao, F. Luo, and X. D. Gu. Greedy routing with guaranteed delivery using ricci flows. In *Proc. of the 8th International Symposium on Information Processing in Sensor Networks (IPSN'09)*, pages 97–108, April 2009.

[20] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1997.

[21] P. F. Syverson, M. G. Reed, and D. M. Goldschlag. Onion routing access configurations. In *DISCEX 2000: Proceedings of DARPA Information Survivability Conference and Exposition*, pages 34–40, January 2000.

[22] W. Trappe and L. C. Washington. *Introduction to Cryptography with Coding Theory*. Prentice Hall, 2002.

[23] Y. Xi, L. Schwiebert, and W. Shi. Preserving source location privacy in monitoring-based wireless sensor networks. *Proceedings 20th IEEE IPDPS*, 06:1–8, 2006.