Effective Probabilistic Approach Protecting Sensor Traffic

Xiaoyan Hong*, Pu Wang†, Jiejun Kong‡, Qunwei Zheng*, Jun Liu*
* Computer Science Department, University of Alabama, Tuscaloosa, AL 35487
† Mathematics Department, University of Alabama, Tuscaloosa, AL 35487
‡ Computer Science Department,University of California, Los Angeles, CA 90095

***Abstract*— Sensor networks are often deployed in environments where malicious nodes present. Among all possible forms of the attacks threatening the sensor networks, in this work, we focus on traffic analysis attacks. Typically, in performing traffic analysis, an attacker will eavesdrop on-going wireless transmissions and analyze contents and timing instances of the transmissions so to infer critical events or to trace valuable assets in the network (e.g, data sources or sinks). The paper presents a probabilistic approach to shape the sensor network traffic to decorrelate time instances in transmissions. The security properties of the approach are studied both analytically and empirically, showing strong protection in high probability.**

I. Introduction

Rapid advances of wireless sensor network technologies have enabled numerous applications in civilian, law enforcement and military environments, from large scale environmental monitoring, structural defect monitoring, border intrusion tracking, hostile terrain surveillance, to small scale personal health monitoring. Many applications of wireless sensor networks require strong security protection over the network and over the data collected and transmitted in the network. Among various security threats, traffic analysis is one that has been made easier in the wireless medium than the wired lines, in that a malicious node can eavesdrop all the wireless transmissions without physically attaching to a line or compromising a node. Though the wireless communications can be protected by strong cryptographic methods at application (end-to-end) or at link level, an eavesdropper can still obtain information on traffic pattern, traffic change pattern, or end-to-end traffic flow paths [8][9]. With such information, the eavesdropper can infer the events happening in the network, or trace the paths to the sources/destinations of the flows or the locations of the events. More deadly consequences could directly result from the information. Thus countermeasures that prevent sensor traffic against traffic analysis have to be studied as an important sensor security issue.

Traffic analysis [1] can be performed on both packet contents (content analysis) and timing events of transmissions (timing analysis). Typical countermeasures to content analysis are padding data packet into a constant length and using per hop key to encrypt packets. When source and destination addresses in packet headers are concerned, per hop link level encryption or link pseudonyme can be used [11]. With these methods, every transmission looks different to an observer. Though encryption and decryption is considered expensive with regard to a sensor's resource constraints, measurements have shown that the overhead is tolerable, e.g., the times that a Berkeley MICA1 Mote sensor takes to encrypt and decrypt 30 bytes data using RC5 are 1.94 and 2.02 msec respectively [6][7]. In addition, encryption serves a broad range of security purposes like data integrity, content privacy, access authentication, etc.

Timing analysis [15] can not be prevented with the encryption methods. Successful correlation between two transmission events reveals the path of a data flow. Solutions studied for wired networks [2][4][14][17] use traffic mixing techniques including sending messages in reordered batches, sending decoy/dummy messages, and introducing random delays. However, sensor networks have features different from a wired network, requiring reevaluation of the existing work and new designs of anti timing analysis solutions.

Features of wireless sensor networks include constraints on the resources each sensor possesses, namely, computing, storage, bandwidth and energy. Such constraints raise concerns on a scheme that requires decoy packets due to the extra usage of bandwidth and energy. On the other hand, communications among sensors is multiplexed within a radio range (depending on encryption method) in the wireless medium, which helps hiding a particular transmission among neighbors' transmissions. Our study, thus, adopts the random delay approach instead of using decoy packets and exploits the multiplexing feature of the wireless medium to protect sensor traffic from timing analysis.

We base our work on the assumption that a sender is able to hide the identity of its intended receiver in transmission but not its own. The rationales behind this assumption are that: on one hand, the sender can either use link level encryption or simply broadcast its messages to hide the receiver. The intended receiver shall use the right key to decrypt the messages; on the other hand, an eavesdropper assisted with radio detection technique can detect a wireless transmission from a referable location and name it. By discovering the temporal dependency of two transmissions at different locations, he acquires the knowledge about the relationship of two consecutive nodes along a data path. When attackers have substantial network-wide monitoring ability, or physical mobile tracking ability, they will be able to integrate piecewise information into global knowledge to finally infer the locations of the events or the base stations in the sensor network.

In this paper, we study random delay strategies to prevent temporal correlation of wireless transmissions within a neighborhood. Specifically, our schemes introduce random delays independently and distributively at each hop. Our strategies do not use dummy messages but exploit the broadcast media in improving the effectiveness and the efficiency of the schemes. The key of such design is how to derive the delay and what the anonymity properties will be. We introduce two schemes and analyze their anonymity properties in the paper. The work

is presented as follows. We start with brief reviews on related terminologies and previous work in Section II, follow by the descriptions of the schemes in Section III. Section IV analyzes the security properties of the schemes, and Section V validates them through simulations. Section VI concludes the paper with future work outlined.

II. Background and Related Work

A. Terminology

The concept of *anonymity* is defined as "the state of being not identifiable within a set of subjects, the *anonymity set*" [13]. The anonymity set is the set of the identities of possible senders/recipients [5][16]. The larger the set is, the better the anonymity protection will be. Further, anonymity is defined in terms of the relationship of a sender and a receiver not to be identified. If an algorithm always provides the size of the anonymity set to be greater than 1, the algorithm provides *deterministic anonymity*. Otherwise, if an algorithm has a probability that the size of the set goes to 0 in an exponentially decreasing rate given a linear increasing complexity parameter of the algorithm, the algorithm is said to provide *probabilistic anonymity* [10].

B. Related work

Strategies thwarting timing analysis are studied within many Internet anonymity protocols, e.g., with various MIX-Net designs [3][10][14]. The strategies include introducing random delay; buffering messages in a pool, reordering messages and flushing the pool; injecting dummy/decoy packets; and padding each packet to the same length or random lengths. For example, a delay-and-playout technique (traffic mixing) is used in [2][14]. Serjantov et al. [17] classify traffic mixing strategies into two major categories—simple MIXes and pool MIXes. Both categories use either one threshold or both: a message pool size n and/or a time period t that a message stays in the pool. Decoy messages could be used when necessary. These threshold based schemes are vulnerable to either flooding attacks, where the adversary sends its own n-1 messages to flush a pool of size n; or trickle attacks, where the adversary blocks all the incoming messages except the target one until the mix node sends it out after t; or the two blended. Various mixing schemes have been proposed to reduce the success ratio of such attacks [17].

A close related work is Stop-and-Go-MIX (SG-MIX) [10]. SG-MIX does not use thresholds on a message pool or a time window, nor decoy messages. Rather, it uses independent random delays for each message. In SG-MIX, the source pre-computes a route passing through a few MIX nodes and values of random delays at each node. The values are encrypted by the keys of each node en route and embedded in the message header. SG-MIX achieves *probabilistic anonymity*, i.e., two transmissions could be correlated with an exponentially decreasing probability when traffic intensity increases linearly. However, SG-MIX can not be used in wireless sensor networks due to many reasons: (i) The ene-to-end approach is not suitable for distributed sensor networks; (ii) pre computed data paths can not be obtained in sensor networks; (iii) cryptographic keys are more likely managed locally in sensor networks; (iv) many applications use mobile sensor networks; and moreover, (v) while per flow treatment by SG-MIX is not a problem for a wired line because MIXes usually have high traffic volume, inputs to a wireless sensor node is not as high as those to a wired MIX. Our schemes use a distributed approach and exploit the broadcast nature of the wireless medium to facilitate anti-traffic analysis strategies.

For wireless multihop networks, preventing traffic analysis has been studied as a routing problem. Jiang [9] presented a routing algorithm that finds appropriate routing paths for various end-to-end flows aiming at maintaining global link traffic patterns as invariable as possible. Our work differs from the work in that we model aggregated traffic in a local transmission radius. For sensor networks, anti-traffic analysis strategies have also been presented in [6], where window based delay randomization and dummy packets are used. Our approach does not use dummy packets, rather, longer delays in trading for bandwidth. By exploiting the wireless media, we are able to control the overall average delay.

III. Probabilistic Traffic Shaping

In this section, we describe two random delay strategies to shape data traffic so that direct correlation of a previous transmission (by the upstream node) with the current one is impossible with high probability. Our strategies do not use dummy messages but exploit the broadcast media in improving the effectiveness and the efficiency of the schemes. Analysis of the anonymity properties will be given in the next section.

A. Network Model

In our targeted sensor network scenario, all the sensors independently generate data reports and relay reports for some of its neighbors. The packets are padded to a constant size. We assume all the sensors have a radio range of r. Sensors within the range r can either send to or receiver from each other. Our assumptions on sensor security follow the ones used by [6][12], i.e., sensors have established pair-wise keys to encrypt transmissions at each hop and also to decrypt the packets and process them at the intended receivers within the transmission range. A further relay of the message will be re-encrypted to generate a different payload pattern. Note this network security model is vulnerable to localized DOS attacks (energy depletion) but the damage will not propagate further.

Using current radio detection technique, an eavesdropper can easily detect a wireless transmission from a referable location. It can identify this transmission using its own coordination system and name convention. When it intercepts two consecutive transmissions from two sensors that are within the range r, the eavesdropper can perform traffic analysis on the two packets for content correlation and timing correlation. With the above sensor security assumption, content correlation is impossible (is hard or resource consuming). Because a relay node will re-encrypt the content with a different key, the same data payload appears differently at every relay transmission. However timing correlation is possible if one packet is sent after a reasonable delay counting the processing

and media access control. More over, a compromised node is more dangerous in that it can act protocol-compliantly but to break the privacy and security system. For example, it can generate its own payload and monitor who will relay the packet. When an adversary detects the relation between upstream and down stream nodes, it can trace the routing path towards either the data source or the destination (base station) by physical movements or by feeding information to its global adversarial information center for integration with other data. These actions could lead to deadly consequences.

B. Distributed Random Delay Strategies

Communications in wireless sensor networks typically lack global centralized control and ene-to-end knowledge. MIX-net techniques, including SG-MIX, can not be used directly. Our attempt is to apply the end-to-end SG-MIX scheme in a distributed way, i.e., each node along a path independently introduces a random delay for the packet it relays. Thus we are able to avoid the many drawbacks of original SG-MIX.

Our first scheme thus works as follows. In the scheme, a sensor node independently generates data packets and emits them immediately after generation. A sensor also relays packets for its neighbors. When a packet is received, it will be delayed for a while before being emitted again. The value of the delay draws from an exponential distribution with parameter μ. When a transmission is delayed, the packet is put into a queue until the delay time expires. During the delay period or even before receiving the packet, the neighbors of the node may transmit or have transmitted packets, or the node itself may have originated packets. Thus when this node transmits the delayed packet, an eavesdropper may not be able to tell whether the current transmission is an originated data or a relay of an early transmission; or, to which early transmission this packet might relate. Thus a relay transmission is hidden among its own data, neighbors's transmissions or its other relay traffic.

The scheme presents the worst case when a packet is emitted after delay but with no mixing traffic. When this happens, an eavesdropper has the highest probability (50%) in guessing whether this transmission is a relay of the previous one or an emission by the node itself. If an eavesdropper is able to identify idle periods of the system, especially, when traffic load is not high, he will have a higher probability in correlating events (we defer analysis to the next section). To be more specific, let's define the following events.

- Event A: the queue is empty when the packet first arrivals at the node;
- Event B: During the delay period, there is no other transmissions in the neighborhood.
- Event C: During the delay period, there is no packets originated by the node itself.

Thus at the time of an expiration, if at least one of the three events does not occur, immediate transmission of the current packet is considered safe because it is mixed with other transmissions when observed by another node in the neighborhood. However, when all of the events A, B and C have happened, transmission of the packet is less safe because this is the only transmission following the previous transmission by a neighbor. We will show that such probability is very low.

We extend the above scheme to further eliminate the worst case by introducing a second or more delay(s) if each expiration of delay returns finding event A happened at the first place and all the later delays have both events B and C happened. There are several issues relate to this extension. First, this could lead to long total delay depending on traffic load. We argue that since a source always sends its packets without delaying, any long delay will eventually end when any of its neighbors or itself sends a packet. The occurrence of more than one delays has decreasing smaller probabilities. Second, we introduce a delay upper bound D_{max} at each hop. The upper bound enables the control over the overall end-to-end delay to meet QoS requirements. Especially, if the sensor network is delivering time critical data, the upper bound provides a tradeoff for optimal anonymity control and performance. Finally, this method creates more chances for a malicious node to perform trickle (or blocking) and flood attack - only one packet is enough to trigger an emission from its neighborhood. When this happens, this scheme reverts to the previous one. However, a successful trickle and flood attack has also very small probability in success, given that sensor sources are uniformly distributed and each source generates packets independently.

We name the basic scheme *probabilistic reshaping (PRESH)* and the extended one *extended probabilistic reshaping (exPRESH)*. In the next section, we will provide analysis on the security properties of the two schemes. The distributions enable a better understanding, especially about the trade-offs between delay and protection.

IV. Security Analysis

A. Attacks

An eavesdropper in the system running PRESH or exPRESH has no predictable clues in correlating two transmissions since each packet is delayed independently and randomly. Transmission events $\{e_1, e_2, ..., e_h\}$ that the eavesdropper has collected during a time window T_h do not help him in guessing which event the next transmission will most likely relate to. Each event has the same probability of $\frac{1}{h}$. However, if an eavesdropper is able to monitor the neighborhood for a long time, he might be able to identify the busy period and the idle period of the system, especially, when traffic load is not high. Thus, when a packet enters the system with an empty queue, and when it is emitted again but there is no other transmissions in the neighborhood up to the time, the eavesdropper can correlate the two events most successfully. This presents the worst case for the two schemes.

In the wireless medium, a single compromised node can only block traffic passing through itself. To successfully break the mixing scheme supported by the neighborhood, the adversary has to compromise a number of nodes in the neighborhood. These nodes then act coordinately to block the transmissions in the neighborhood except the target packet for a period in order to empty the queue and to wait for the target packet to come out. If the adversary could obtain D_{max}, he can block the neighborhood for the same period to

increase blocking success probability. Then exPRESH reverts to a threshold-based scheme. The success rate of such attack depends on the time period the adversary spends in blocking the traffic. After that the adversary only needs to send one additional packet to convince the target packet to come out after its delay period. In this case, exPRESH reverts to PRESH. After all, launching such attacks requires very strong adversary.

B. Assumptions

For the convenience of the analysis, we assume transmission delay and processing delay are negligible compared to the random delay we introduced. Benchmark measurements of Berkeley Mote MICA sensors show that transmission of a sensor data packet uses 40-50 ms, encryption/decryption requires about 2 ms using RC5 for 30 bytes data, and sensors generate data at a rate of 60s [6]. Thus it is safe to assume a mean of random delay at 60s or tens seconds. The aggregated transmissions over the neighborhood, provide enough mixing traffic.

We base our analysis on the uniformity of node distribution and traffic distribution. In a neighborhood with m nodes, each node originates data independently and each has equal probability in relaying data for its neighbors. We simplify the problem by assuming a packet flow will be relayed in a neighborhood only once. We also simplify the model by not differentiating a specific node from its neighbors since we study the worst case. Thus, we model the traffic that each node generates (both originating and relaying) as a Poisson process with parameter λ. The aggregated traffic in a neighborhood is a Poisson distribution with parameter $m\lambda$ (including the node itself). At a specific node, a portion of the arrivals (at rate of λ) will be processed (here, delayed and then re-emitted) with a service time exponentially distributed with parameter μ immediately at arrival. From the view point of an eavesdropper, the system we are modelling is the neighborhood around it. So we redefine the events that lead to the worst case, i.e., a transmission of a packet directly following its previous transmission, to the following:

- Event A': the queue is empty when the packet arrivals at a node;
- Event B': during one delay period, there is no transmissions in the neighborhood.

Based on this aggregated neighborhood traffic model, we analyze the anonymity properties for both PRESH and exPRESH schemes.

C. PRESH

For PRESH, a packet will be delayed only once. We model the scheme as a $m/m/\infty$ queue system, where the arrivals are in a Poisson distribution with parameter $m\lambda$, and infinite number of servers are able to serve each arrival immediately with a service time exponential distributed with parameter μ. With this model, the analysis for PRESH follows the work presented in [10] directly. We give the main results below with a definition on traffic intensity $\rho = m\lambda/\mu$.

1) Probability of worst case: The worst case happens when the two events A' and B' occur. The probability is

$$P(worstCase) = P(A' \cap B') = P(A')P(B')$$

For event A', $P(A')$ is simply the probability that the system is empty at an arrival, which is $e^{-\frac{m\lambda}{\mu}}$. For event B', the probability that during a service time there is no arrival is equal to the probability that an sample drawn from $Exp(\mu)$ is larger than a sample from $Exp(m\lambda)$, which is $\frac{\mu}{m\lambda+\mu}$. Thus, we have

$$P(worstCase) = \frac{e^{-\rho}}{1+\rho} \tag{1}$$

2) Size of anonymity set: When a packet is emitted, the packet is mixed within the anonymity set. The anonymity set of a packet X consists of the packets in the queue when X arrivals, X itself and the packets arriving during the service time of X. The expectation of the number in the queue when a packet arrives is $\frac{m\lambda}{\mu}$. And the expectation of the number arrivals during the busy period of a packet is $\frac{e^{\frac{m\lambda}{\mu}}+1}{2}$. Thus the expectation of the size of the anonymity set U_0 is:

$$U_0 = \rho + \frac{e^{\rho}+1}{2} \tag{2}$$

Note that $\lim_{\rho\to 0} U_0 = 1$, i.e, when traffic load is very low, the only transmission would be the packet itself. At that time, the scheme fails to protect X by mixing it with others: $\lim_{\rho\to 0} P(worstCase) = 1$.

3) Success rate of blocking attack: Suppose a time interval τ is used by the adversary to block the traffic and then to send its own one packet. The blocking attack will success if all the packets in the system leave within time τ. The memoryless property gives us the probability that a packet in the system leaving within τ as $P[X \le \tau] = 1 - e^{-\mu\tau}$. At the time the blocking attack starts, the system could have arbitrary number of packets, say i, with a probability of $P[X = i] = \frac{\rho^i}{i!}e^{-\rho}$. The probability that all of them leave the system within time τ is $V[allLeave|X = i] = (P[X \le \tau])^i = (1-e^{-\mu\tau})^i$. Thus, the expectation of success probability of blocking attack is

$$V_0(\tau) = \sum_{i=0}^{\infty} P[X=i](P[X\le\tau])^i = \sum_{i=0}^{\infty} \frac{\rho^i e^{-\rho}(1-e^{-\mu\tau})^i}{i!}$$

$$= exp(\frac{-m\lambda e^{-\mu\tau}}{\mu}) \tag{3}$$

Equation (3) shows that if a blocking interval τ is set, the success probability decreases exponentially with the linear decrease of the delay parameter μ.

D. exPRESH

For exPRESH, additional delays will be used when PRESH's worst case occurs with a probability of $P(exPRESH) = P(worstCase)$. When that happens, an independent exponential delay starts. The procedure repeats thereafter if the message terminates its service but finds there is no arrival to the system during its last service time. The procedure stops until at least one new message

has arrived or the maximum delay bound D_{max} has been reached, then the packet is emitted. When the scheme is under blocking attack, the best strategy the attacker will use is to force a packet to come out after the first delay. In that case, exPRESH reverts to PRESH. The key performance issue for exPRESH is the trade off between a longer delay and a zero probability of worst case. The total delay time a packet experiences becomes a critical factor. We model the system using the state transition diagram shown in Figure 1. In this model, the bound is not considered.

In the diagram, $n, n \geq 0,$ denotes the states of queue length n. States $1F$ and $1L$ are the steady states for the first and last message during a busy period in the system. The corresponding probabilities of queue length in steady-states are x_n. x_{1L} and x_{1F} are for steady states $1F$ and $1L$ respectively. We have the following balance equations:

$$\begin{aligned} m\lambda x_0 &= \mu x_{1L} \\ m\lambda x_{1F} &= m\lambda x_0 \\ (m\lambda+\mu)x_{1L} &= 2\mu x_2 \\ (m\lambda+2\mu)x_2 &= m\lambda x_{1F} + m\lambda x_{1L} + 3\mu x_3 \\ &\cdots \\ (m\lambda+n\mu)x_n &= m\lambda x_{n-1} + (n+1)\mu x_{n+1},\ n>2 \end{aligned}$$

The derivation of x_n is given in the Appendix I. An explicit expression for x_0 is

$$x_0 = \frac{e^{-\rho}\rho}{(1+\rho)-e^{-\rho}} \qquad (4)$$

Note that $\lim_{\rho\to 0} x_0 = \frac{1}{2}$. Based on the probabilities, we are able to give the following properties.

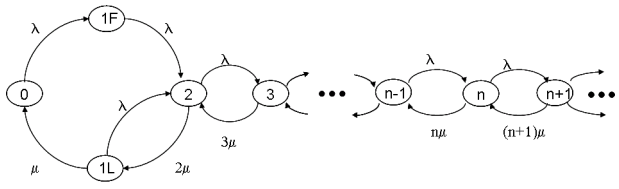

Fig. 1. State Transition Diagram

1) Delay Distribution: For exPRESH, a packet may be delayed for many terms till there is at least one arrival (packet M) during its last delay term. Thus the total period D a packet is delayed (the busy period) can be divided into two sub periods T_1 and T_2, where T_1 is the time for the first packet M to arrival in a busy period, and T_2 is the residual service time after M arrivals. $D = T_1 + T_2$. Due to the memoryless property of the arrival and delay processes, T_1 has the exponential interarrival distribution with parameter $m\lambda$, and T_2 has the exponential service time distribution with parameter μ. The probability density function of D is:

$$\begin{aligned} f_D(t) &= \int_0^t m\lambda e^{-m\lambda s}\cdot \mu e^{-\mu(t-s)}\, ds \\ &= \begin{cases} \frac{m\lambda\mu e^{-\mu t}}{m\lambda-\mu}(1-e^{-(m\lambda-\mu)t}), & if\ m\lambda \neq \mu \qquad (5) \\ m\lambda\mu t e^{-\mu t}, & if\ m\lambda = \mu \qquad (6) \end{cases} \end{aligned}$$

Note that when $m\lambda = \mu$, $f_D(t)$ becomes a $Gamma$ distribution with parameters $m\lambda$ and 2. Figure 2 illustrates the distribution of the total delay needed for a successful mixing with $\mu = 1\, packet/second$. A reference curve of $Exp(\mu)$ for PRESH is given too. The figure shows that when $\rho = 5$, exPRESH and PRESH have very close probabilities at long delay times. Owing to neighborhood multiplexing, $\rho = 5$ is easy to reach. For example, for a neighborhood with 10 active members, the mean delay could set to only a half of the mean packet interarrival time.

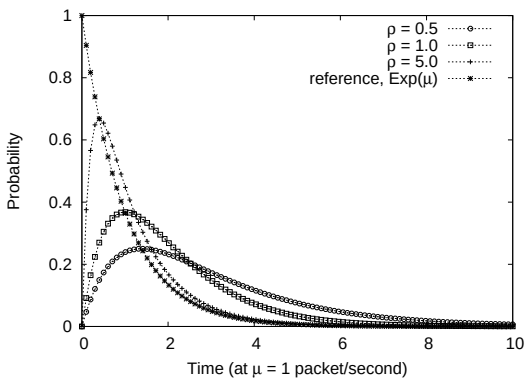

Fig. 2. Distribution of Delay

2) Mean total delay time: Now we compute the mean of the total delay for exPRESH. The mean total delay time for any message can be computed by conditioning on the system is empty or not when the message arrives at the system. When the system is empty, the massage has to wait until one arrives. Thus,

$$D_1 = x_0\left(\frac{1}{m\lambda}+\frac{1}{\mu}\right) + (1-x_0)\frac{1}{\mu} = \frac{1}{\mu}+\frac{x_0}{m\lambda}$$

Substituting x_0 yields the mean total delay time

$$D_1 = \frac{1}{\mu}\left(1+\frac{e^{-\rho}}{(1+\rho)-e^{-\rho}}\right) \qquad (7)$$

One can see that when the traffic intensity ρ is large, the mean total delay is just the mean service time μ^{-1}. When ρ is small, however, the total delay increases because the first message in the busy period has to wait for a new one to come. The additional delay $x_0/m\lambda$ could be substantially large. An illustration of the function is given in Figure 3, where the mean total delay is normalized to the mean service time. In order to demonstrate the convergence to 1, X axis starts at 0.25 in the figure.

V. Simulations

We investigate the influence of aggregated traffic on the mixing schemes PRESH and exPRESH through simulation. We use GlomoSim[18], a packet level simulator for wireless and wired networks. In the simulation, 225 nodes are uniformly distributed in a square field at an average nodal density of 16 nodes per transmission radius. In an attempt to collect data from uniform traffic pattern, we set sinks on one side of the field and all the data flow towards the same direction. The statistics are only collected from the nodes at the other side of the field. Those nodes also have their neighborhood within the

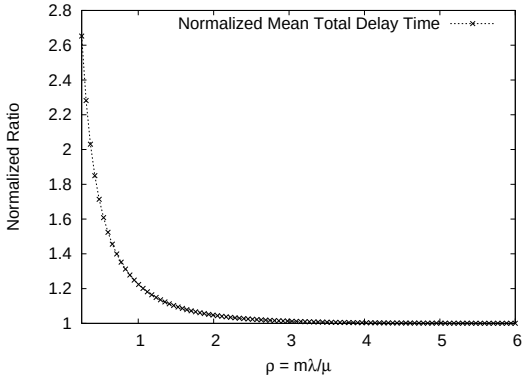

Fig. 3. Mean Total Delay Time Normalized to μ

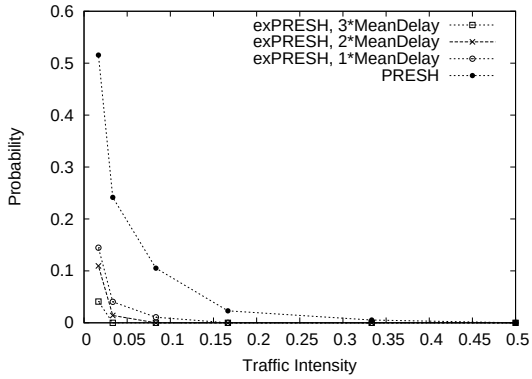

Fig. 4. Probability of Worst Case

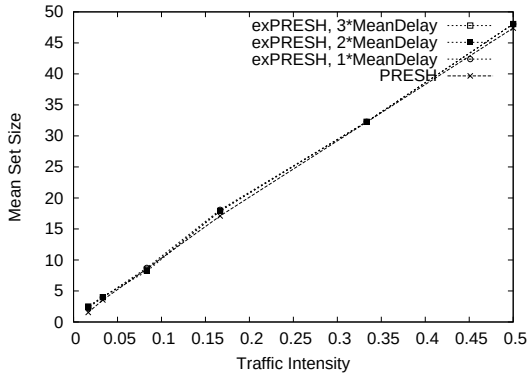

Fig. 5. Size of Anonymity Set

simulation area. Even though, for the simulations conducted, we can not enforce that one flow being relayed only once in a node's neighborhood. We run the simulations over multiple seeds for the random number generator. The results should reflect accuracy statistically. In simulations, we set the mean date generating interval to be fixed at one report every 60 seconds [6]. The mean delay time varies from one second to 60 seconds. At short delay intervals like one second, a node does not have enough aggregated transmissions to have at least one packet on average in order to hide its own traffic. The X axis is the relay traffic intensity ρ_0 for one node, i.e, λ/μ. Thus when each node generates packets every 60s on average in a neighborhood of 15 nodes, a node has to wait for 4 seconds ($\rho_0 = 0.067$) on average to hear another transmission. If a node delays a packet longer, it has better chance for mixing.

We collect statistics to calculate average $P(worstCase)$ for PRESH and the size of the anonymity set for both schemes. For exPRESH scheme, the worst case occurs when the total of extended delays exceeds the bound D_{max}. At that time, even without an arrival, the packet has to be emitted. We count the occurrences of this event to calculate the $P(worstCase)$ for exPRESH. Obviously the probability is affected by the value of D_{max}. In the simulation, we vary D_{max} to be one, two or three times of the chosen mean delay for exPRESH.

Figure 4 gives the probabilities for the worst cases. The possibilities approach zero quickly when the traffic intensity increases. Clearly exPRESH schemes decrease faster than PRESH. In addition, the larger the delay bound, the smaller the probabilities. The result suggests that we can trade delay for stronger security protection. Figure 5 gives the sizes of the anonymity sets corresponding to the above configured parameters. The sizes increase when load increases. The two schemes demonstrate little differences due to the fact that most packets experienced the same neighborhood activities. Differences exist when load is low.

The distribution of the total delays that packets experienced at each node is illustrated in Figure 6 and 7 with mean delay setting to one second and 60 seconds respectively. For both figures, the bound D_{max} is enforced in such a way that a delay drawn from the $Exp()$ must be finished even if D_{max} is met. This allows us to show the full distribution smoothly. Otherwise, one can expect the curves end with a peak at the corresponding D_{max} values. When calculating probability, the placement of bins affects the shape of the curves slightly. Here, each figure takes 20 bins. The corresponding value is drawn at the right side of a bin. For example, a point drawn at "Per Hop Delay = 1s" in Figure 6, counts data fallen in the bin of $(0.5, 1]$. This particular case actually counts for the leverage of a possible higher peak at "x = 1s".

In Figure 6, PRESH shows exponential decay trend of the probability with regard to the delay time, which is expected. For exPRESH, more terms of delays will be used since the probability of the worst case occurring after the first delay is relatively high at one second as the mean delay time. The figure shows that exPRESH exhibits higher probabilities of longer overall delays than those of PRESH. The curves shift towards the right side of PRESH. Especially, larger D_{max} shows more shifting corresponding to longer delays. When the mean delay reaches 60 second (Figure 7), all the schemes reveal the same distribution. This is because that the aggregated traffic intensity is so high that all the packets will be transmitted after one delay no matter what scheme is in use. The distribution converges to the exponential distribution with μ.

VI. Conclusion and Future Work

The paper has presented traffic shaping schemes to countermeasure traffic analysis over sensor data transmissions. PRESH and exPRESH both use random delays to de-correlate transmission events without introducing extra bandwidth overhead for sensors. The approaches also exploit the traffic multiplexing ability of the wireless medium to reduce the overall delay and to reduce the probability of the worst case. exPRESH presents an improvement over PRESH to eliminate the occurrence of the worst case at the cost of longer ene-to-end delays. Analysis on the anonymity properties and simu-

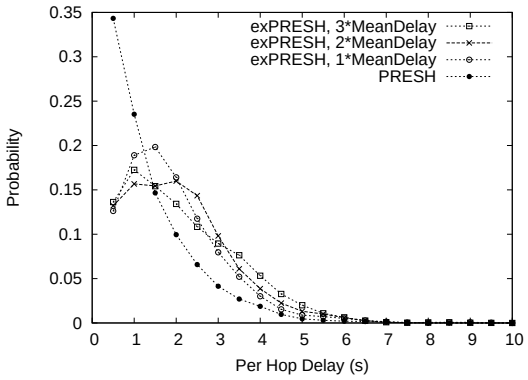

Fig. 6. Distribution of Delay at Mean = 1 second

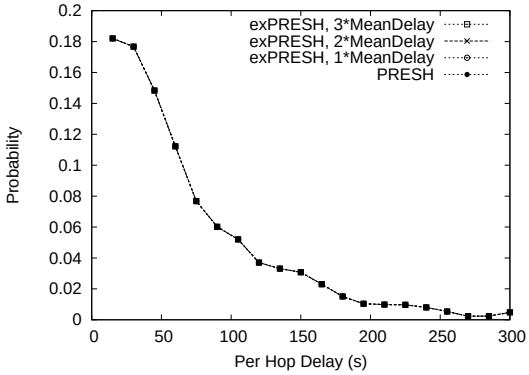

Fig. 7. Distribution of Delay at Mean = 60 seconds

lations are given in the paper. The main results are that with very high probability, the schemes successfully mix sensor data transmissions among neighborhood radio activities; and the insecure facts of the schemes decrease exponentially with linear increase of mean delay time.

Some applications of sensor networks do not demonstrate uniform traffic patterns, e.g, applications with only a few base stations will show traffic concentration near the base stations. Our future work will study traffic shaping strategies for these traffic patterns. We will also analyze the impact of capacity on neighborhood mixing technique.

REFERENCES

[1] A. Back, U. Möller, and A. Stiglic. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems. In I. S. Moskowitz, editor, *Fourth Internationl Workshop on Information Hiding (IH'01), Lecture Notes in Computer Science, 2137*, pages 245–257, 2001.

[2] O. Berthold, H. Federrath, and M. Köhntopp. Project Anonymity and Unobservability in the Internet. In *Computers Freedom and Privacy Conference 2000 (CFP 2000), Workshop on Freedom and Privacy by Design*, 2000.

[3] O. Berthold, H. Federrath, and S. Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 115–129, 2000.

[4] O. Berthold and H. Langos. Dummy Traffic against Long Term Intersection Attacks. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002), Lecture Notes in Computer Science 2482*, pages 110–128, 2002.

[5] D. L. Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[6] J. Deng, R. Han, and S. Mishra. Intrusion Tolerance and Anti-Traffic Analysis Strategies for Wireless Sensor Networks. In *IEEE International Conference on Dependable Systems and Networks (DSN)*, pages 594–603, 2004.

[7] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu. Analyzing and Modeling Encryption Overhead for Sensor Network Nodes. In *the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, Sept. 2003.

[8] A. Hintz. Fingerprinting Websites Using Traffic Analysis. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002), Lecture Notes in Computer Science 2482*, pages 171–178, 2002.

[9] S. Jiang, N. Vaidya, and W. Zhao. Routing in Packet Radio Networks to Prevent Traffic Analysis. In *IEEE Information Assurance and Security Workshop*, 2000.

[10] D. Kesdogan, J. Egner, and R. Buschkes. Stop-and-go MIXes Providing Probabilistic Security in an Open System. *Second International Workshop on Information Hiding (IH'98), Lecture Notes in Computer Science 1525*, pages 83–98, 1998.

[11] J. Kong and X. Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In *ACM MOBIHOC'03*, pages 291–302, 2003.

[12] C. Ozturk, Y. Zhang, and W. Trappe. Source-Location Privacy in Energy-Constrained Sensor Network Routing. In *ACM SASN*, pages 88–93, 2004.

[13] A. Pfitzmann and M. Köhntopp. Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 1–9, 2000.

[14] A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDNMixes: Untraceable Communication with Very Small Bandwidth Overhead. In *GI/ITG Conference: Communication in Distributed Systems*, pages 451–463, 1991.

[15] J.-F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *DIAU'00, Lecture Notes in Computer Science 2009*, pages 10–29, 2000.

[16] A. Serjantov and G. Danezis. Towards an Information Theoretic Metric for Anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002), Lecture Notes in Computer Science 2482*, pages 41–53, 2002.

[17] A. Serjantov, R. Dingledine, and P. F. Syverson. From a Trickle to a Flood: Active Attacks on Several Mix Types. In F. A. P. Petitcolas, editor, *Fifth International Workshop on Information Hiding (IH'02), Lecture Notes in Computer Science, 2578*, pages 36–52, 2002.

[18] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory. GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems. http://pcl.cs.ucla.edu/projects/glomosim/.

APPENDIX I
STEADY-STATE PROBABILITY x_n

The balance equations can be written as:

$$m\lambda x_n = (n+1)\mu x_{n+1},\ n \geq 2$$

Let $\rho = m\lambda/\mu$, we have x_n as functions of x_2:

$$x_n = 2\frac{\rho^{n-2}}{n!}x_2,\ n \geq 2$$

We then obtain the following:

$$x_2 = \frac{\rho(1+\rho)}{2}x_0$$

$$x_{1L} = \rho x_0,\ \ x_{1F} = x_0$$

Using the normalization condition yields

$$x_0 + x_{1L} + x_{1F} + \sum_{n=2}^{\infty} x_n = 1$$

$$x_0\left[2 + \rho + \rho\,(1+\rho)\sum_{n=2}^{\infty}\frac{\rho^{n-2}}{n!}\right] = 1$$

So

$$x_0 = \frac{\rho}{(1+\rho)\,e^{\rho} - 1}$$