Source Capture Time Analysis of Privacy Communication Protocols for Wireless Sensor Networks

Pengjun Pan **Rajendra V. Boppana**
Computer Science Department, UT San Antonio
ppan@cs.utsa.edu boppana@cs.utsa.edu

Abstract— **Communication privacy techniques that protect the locations of source sensor nodes and sink nodes from either global or local adversaries have received significant attention recently. The improvement in capture time, which is the time it takes for an adversary to identify the location of the source, is often estimated using simulations. In this paper, we present probabilistic models to analyze the expected capture time of source sensor nodes for recently privacy protocols against (1) local adversary using phantom routing, and (2) global adversary using statistically strong source anonymity (SSA). Using these models, we show that both phantom routing and SSA fall short of achieving high degree of anonymity. The phantom source routing improves the capture time over normal routing methods, but falls short of possible upper bounds on expected capture time. SSA is prone to simple time correlation attack by a global adversary; even with a large number of dummy message transmitted by non-source nodes, the propagation of one real message from source to sink will be sufficient to identify the source with almost no false positives. Based on these findings, we suggest a simple modification to SSA that increases the false positive rate to nearly 90 percent and thus the capture time.**

I. Introduction

Communication privacy in wireless sensor networks (WSNs) has received significant attention in recent years. The motivation is to protect the source sensor nodes, the nodes that generate messages whenever events of interest occur or tagged valuable assets are in close proximity, or the destinations of the messages (usually, the sinks). Even when the message confidentiality and integrity are ensured with end-to-end encryption and authentication techniques, an adversary may be able to analyze the traffic flows and identify source sensor nodes and thus capture the tagged assets.

Several protocols that attempt to preserve communication privacy have been proposed to protect against local or global of adversaries. In a local adversary model, the adversary has limited hearing range, which is usually a multiple of sensor nodes' transmission range and covers a small portion of the whole network [1]. In a global adversary model, the adversary can hear any packet transmission over the entire network by either having high gain network equipment or deploying a set of small hostile sensors over the whole network and obtaining transmissions cooperatively [2].

There are two main forms of traffic analyses: message timing and message counting. In the timing analysis attack, the adversary attempts to identify the potential routes, sources and sinks based on the timestamps of transmissions by different sensors. For a packet suspected to be carrying a real message, an adversary can determine not only the location of sender using triangulation techniques but also that of the receiver because the receiver will react quickly to the packet it receives, e.g. forwarding further, in most of the cases. Therefore, the adversary can determine the directions of source and sink nodes. With a local adversary model, several such real events must be observed and analyzed by the adversary to locate the source; a global adversary, with the availability of timestamps of all transmissions, needs just one occurrence of a real event to completely analyze the route from the source to sink. Several papers proposed techniques to mitigate this attack by diversifying routing path in the case of local adversary models or by camouflaging real packets in a sea of dummy packets in the case of a global adversary.

In the message counting attack, the adversary attempts to detect and capture the sink by counting the number of transmissions occurred at each location since the sensors closer to sinks are likely to more real transmissions. This can be effectively countered by having a large number of dummy packet transmissions to ensure that the transmission events by nodes are statistically indistinguishable.

Though several interesting communication protocols to mitigate local and global adversaries have been proposed [3], [1], [4], [5], [6], [7], [8], [2], capture time (CT)—the time taken by the adversary to correctly identify the location of a source node—is mostly estimated

using simulations.

In this paper, we analyze two representative privacy protocols that protect source sensor nodes from either local or global adversary, who can launch attacks via timing or counting analysis, using both probabilistic analysis and simulation proof. The protocol against local adversary to be analyzed is called phantom routing [1], which directs real packets to some fake sources, called *phantom sources*, through random walk and followed by shortest path routing. The protocol against global adversary to be analyzed in called statistically strong source anonymity (SSA) [2].

We develop a probabilistic analysis to determine the expected capture time ($\overline{CT}$) for both phantom routing and SSA. In particular, we show that the capture time is significantly longer than the published simulation analysis and verify our analysis with simulations. For SSA, we show that it, although being claimed to be against message counting analysis, is vulnerable to message timing analysis in order to meet a required end-to-end message delay. A simple message timing analysis is proposed to launch such an attack, through which there are virtually no false positives. Based on these findings, we propose a simple modification to SSA to increase the false positive rate to nearly 90% and, thus, improve the capture time.

The rest of the paper is organized as follows. Section II introduces some of the communication privacy protocols proposed in literature followed by a description of network model, attack models, and security goal of this paper. Section III analyzes capture time of local adversary strategies. Section IV provides capture time analysis of global adversary strategies based on a simple proposed adversary strategy. Simulation results are shown in Section V. A modification of SSA is proposed and proved by simulations as well. Then we concludes the paper in Section VI

II. Background

In this section, we describe recent results on communication privacy protocols and present the network model, the attack models, and the security goals.

A. Related Work

Protection of both source node and sink node privacy are addressed in literature. In this paper, we are interested in the source privacy [1], [2], [3], [4], [6], [8]. We do not address sink privacy [5], [7].

Phantom routing [1] protects source nodes from a local adversary. It first uses a random walk for h_{walk} hops to send the data packet to a node called *phantom* source. From the phantom source, the packet can be sent by either probabilistic flooding or single-path routing to the sink. This is one of the early results on privacy communication in WSNs.

The recently proposed statistically strong source anonymity (SSA) protocol addresses the issue of source privacy under attack by a global adversary, who can monitor the traffic in the entire network [2]. This is achieved by making all sensors in the network send either real of dummy packets according to an exponential distribution. The real packets reporting an event are propagated quickly to sinks by triggering transmission of such events within a specified time limit. The dummy packet injection intervals are adjusted so that the adversary cannot identify source nodes with a statistical analysis of packet transmission intervals.

The idea of dummy packet transmission and maintaining the same transmission rate of all nodes to protect sources from global adversaries are also introduced in [5].

B. Network Model

We assume that sensors are deployed with approximately uniform density in a rectangular field with one or more sinks. There are one or more sources reporting sensed data to the sink. We consider wireless sensor networks in which sensors periodically alternate between active mode, during which a node senses, receives, processes and transmits messages, and sleep mode, during which a node shuts down most of the functionality except the circuitry necessary to retain memory and wake up at the appropriate time. Since most applications require sensors to respond within seconds or minutes of the occurrences of events of interest, having a low duty cycle (percentage of the time a sensor node is active) with sleep time adjusted to satisfy application response time constraints improves the overall network lifetime. Long sleep durations and clock drift can make sensor unsynchronized; this can be mitigated using any of the existing synchronization techniques proposed in the literature [9], [10], [11], [12]. Furthermore, the synchronization techniques are designed such that sensors come out of sleep mode in a staggered manner: nodes that are farthest wake up first followed by their parents (next hop nodes on the route to sinks) next, and so on. The time gap between the wake up time of a node and its parent is often called the *offset*. This pattern ensures that a packet injected by a source can reach sink without encountering any sleeping node [12].

All data payloads are of the same size and are encrypted with suitable nonces at each hop using suitable encryption/dectryption and key management techniques [13], [14], [15]. Therefore, the adversary can neither understand the contents of a payload nor distinguish the same payload transmitted by two different nodes.

C. Privacy Metrics

We use capture time (CT), the time it takes to identify a source correctly, and the number of capture attempts (CAs) as the figures of merit or evaluation criteria for privacy protocols.

A capture attempt is the identification of potential source by the adversary and the physical examination of the location of the node to capture the intended tagged asset or gather additional information of an event occurrence. A capture attempt at a sensor node that did not send a real message is a false positive, which must be minimized.

The capture time is a function of CAs, the true positive rate, and the rate at which the real events are generated. The first two factors are strongly impacted by the anonymous communication protocol and the attack strategy, respectively, while the third factor is application specific and is an independent variable in our analysis.

D. Attack Models

Attacks on source location privacy by either local or global adversaries are considered in this paper. A local adversary (LA) has a radio hearing range equal to a multiple of the radio range of sensor nodes. The LA is equipped to locate the source of any transmission it hears. A common and realistic adversary model considered in literature and used in this paper is the patient adversary (PA) model, similar to algorithm 1 in [1], [16]. It is assumed that PA does not know a priori the locations of sensor nodes and that PA starts at the sink, and moves towards a sensor node upon the first hearing of a transmission in the current active period.

A global adversary (GA), used in [2] for example, has the capability to overhear transmissions by all sensor nodes and identify the geographical locations of the senders. Furthermore, the GA has the resources to store the necessary information regarding the transmissions–timestamps, estimated location, etc.–and process them using various methods to identify sources.

In the worst case, the adversary cannot get any clue to move towards the real source. Therefore all the adversary can do is to randomly pick any node in the network and check it. We call this adversary a random adversary (RA). We also define RA-e as a random adversary who eliminates examined node from further consideration. The RA and RA-e are not real adversary models, but they help to analyze CT.

E. Security Goal

A privacy communication protocol should significantly increase the false positive rates of CAs by the adversary to increase its exposure. This also increases the expected source capture time ($\overline{CT}$) for a given event generation rate. We use $\overline{CT}$ as a figure of merit to evaluate communication security protocols.

III. CT ANALYSIS OF LOCAL ADVERSARY STRATEGIES

In this section, we analyze phantom routing proposed by Kamat *et al.* [1], as a typical technique against LA, and present an analytical model to evaluate $\overline{CT}$ for this protocol.

A. Expected capture time

Let RA denote a random adversary who can select one sensor node out of n at random, examine it and, if it is not a source, leave it undisturbed. Therefore, picking the source by RA is a sequence of Bernoulli trials with P(success) $= p = \frac{1}{n}$, where p denotes the probability function and n is the number of sensor nodes. Let X be the random variable to indicate number of capture attempts (nodes suspected and checked) by the adversary before detecting a source node. Then X is a geometric random variable. Therefore, P(X = k) $= (1-p)^{k-1}p, k \geq 1$. The average value taken by X, $\mu_X = \frac{1}{p} = n$. This is the expected capture time. Let RA-e denote the adversary who never picks an already rejected node. Then X is a uniform random variable and $\mu_X = \frac{n+1}{2} \approx \frac{n}{2}$.

Since an RA does not use any intelligent processing of transmission contents and their timings, these bounds may be considered the best possible capture times for the network. Any adversary that uses transmission timing analysis is likely to have lower expected capture time.

B. Application of $\overline{CT}$ bounds

We now estimate the expected capture times by RA and RA-e for the networks considered for phantom routing [1]. These networks are described by the following parameters.

- Network size: $6000 \times 6000\ m^2$
- Number of nodes: 10000
- $h_{walk} = 10$
- $avg_{neighbor} = 8.5$

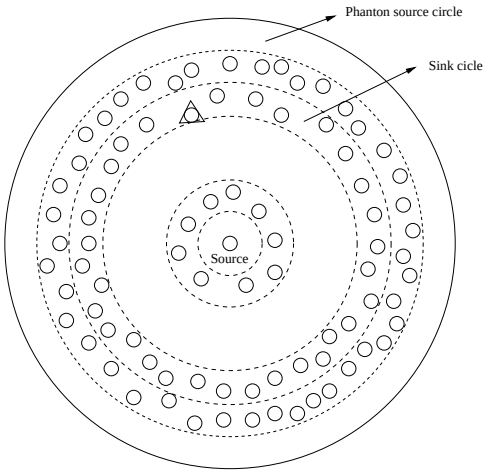

Fig. 1. Phantom routing, case 1

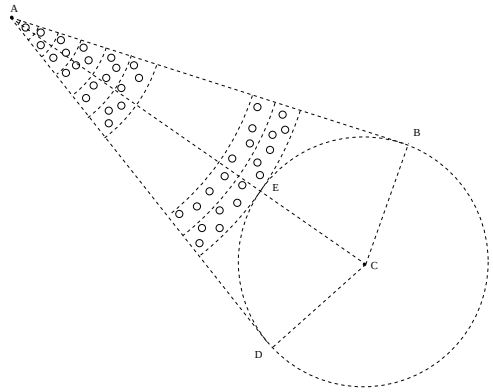

Fig. 2. Phantom routing, case 2

Therefore, the average number of nodes in a radio range is 9.5. If r is the radio range, then $\frac{1000\pi r^2}{6000\times6000} = 9.5$. Solving the equation gives $r = 104$ m.

Two cases were simulated in [1]. In the first case, the sink is 8 hops away from the real source, while it is 34 hops away in the second case. In both cases, the patient adversary (PA) model and single-path routing are used.

For easier description, imagining that sensors are located on different bands centered by the source node based on their hop distances to the source, we define phantom source circle as the circle in which all nodes are h_{walk} hops away from the real source and may potentially be selected as phantom sources with equal probability. Therefore, the first case corresponds to the situation where the adversary is within the phantom source circle; the second case corresponds to the situation where the adversary is outside the phantom source circle. We estimate the expected capture times by RA and RA-e in each case.

1) Adversary is inside the phantom source circle: Figure 1 depicts case 1. The node in the middle is the real source and the triangle represents the RA-e adversary. We define sink circle as the circle in which all nodes have the same hop distance to the real source as the hop distance between the sink and the real source. The outermost circle is phantom source circle. In the worst case, from the network defender's point of view, the RA-e adversary checks only the nodes in the sink circle or nodes closer to the source. There are $\frac{\pi((8\times104)^2)\times10000}{6000\times6000} = 604$ nodes to be examined. So the expected capture time with RA-e is 302.

2) Adversary is outside the phantom source circle: In case 2, shown by Figure 2, the adversary from A, where the sink is located. All the nodes within the indicated circle may be involved in the random walk phase. Since packets are transmitted using single-path routing in the second phase, the nodes that are involved in the transmissions are inside the area ABCD. Similar to the analysis given above, the lowest expected capture time occurs when the adversary checks only the nodes within this cone and the circle.

The area ABED can be calculated by subtracting the sector CBED from the quadrangle ABCD. First of all, we need to calculate the angle $\angle BCD$.

$\angle BCA$ can be measured by knowing the distance $|AC|$ and the radius of the circle, 10. So $\angle BCA = cos^{-1}(\frac{10}{34}) \approx 73°$. Thus we have $\angle BCD = 146°$.

Since $\angle ABC = \angle ADC = 90°$, $|AB| = |AD| = \sqrt{34^2 - 10^2} = 32.5$ hops $= 3380$ m.

Therefore, the number of nodes in region ABED is $\frac{3380\times1040 - \frac{146}{360}\times\pi\times1040^2}{3600} = 593$.

And the number of nodes in the phantom circle is $\frac{\pi\times1040^2}{3600} = 944$.

So an RA-e adversary has an expected capture time of $\frac{593+944}{2} = 769$.

Without phantom routing, the expected capture times for the two cases are 8 cycles and 34 cycles, respectively. The phantom routing improves the expected capture time to 33 cycles and 91 cycles (The reported safety period in [1] are 32 and 90). This means that phantom routing does improve the expected capture time by a RActor of 3 to 4. However, the calculations of expected CT for RA-e indicate that the timing analysis used by a PA works in the adversary's RAvor despite phantom routing.

IV. Capture Time Analysis of Global Adversary Strategies

In this section, we analyze expected source capture time of global adversary, using SSA protocol [2] as an example. To counter against an omnipotent global adversary, SSA requires sensor nodes to transmit two

types of messages: dummy messages based on a Poisson process with λ_d as the message rate or $1/\lambda_d$ as the inter-message delay (IMD), and real messages, which carry real event data, according to a Poisson process with event rate λ_e. To avoid long delays in reporting real events to sinks, sensing of a real event or receiving a real message from a downstream node triggers a real message by the source or next hop, respectively, within some specified time limit or upper bound β. To camouflage these triggered events, care is taken to space dummy event transmissions such that overall IMDs by any node in the WSN is approximately exponentially distributed with parameter λ_d. It is claimed and shown using simulations that, though a global adversary can hear all transmissions and has the computational resources to perform any analysis needed, it is incapable of reliably identifying source locations since statistically all nodes are indistinguishable [2]. In the remainder of the section, we develop a probabilistic analysis technique for a global adversary to estimate $\overline{CT}$.

We focus on the 2β-events, which are the instances of consecutive transmissions by a node within 2β seconds. The 2β-events can be dummy events triggered by a node's dummy packet transmission within 2β seconds of its most recent transmission or real events triggered by real packet transmissions. The probability, p_d, that a dummy packet is sent within 2β of a previous transmission by the node can be calculated using the CDF of the exponential distribution.

$$p_d = 1 - e^{-2\beta\lambda_d} \tag{1}$$

Let X be the random variable (RV) for the time elapsed between a previous transmission by a node and either the occurrence of a real event or the arrival of a real message from a downstream node. Let Y be RV for the time it takes to trigger a real message transmission by this node. Note that X and Y are independent RVs. We are interested in p, the conditional probability $P(X+Y \leq 2\beta | Y \leq \beta)$. This requires the integration of the joint-pdf of X and Y for the range $(0, 2\beta)$. We can also estimate its lower bound as follows.

$$\begin{aligned} p &= P(X+Y \leq 2\beta \,|\, Y \leq \beta) \\ &\geq P(X \leq \beta | Y \leq \beta) \cdot P(Y \leq \beta | Y \leq \beta) \\ &= P(X \leq \beta) \cdot 1, \qquad \text{since Y must be } \leq \beta \\ \therefore\ p &\geq 1 - e^{-\beta\lambda_e} \end{aligned} \tag{2}$$

Thus, p_{real}, the probability that an observed 2β-event, was triggered by a real event is

$$p_{\text{real}} = \frac{p}{p + p_d}. \tag{3}$$

To see the significance of (3), let us consider some of the parameter values used in [2]: $1/\lambda_d = 20$ seconds and $1/\lambda_e = 20$ seconds.

Since β is not specified, we estimate a suitable value for the same. Suppose that the radius of a WSN (with a sink as the center) is 20 hops and that the sink needs to be informed when an event occurs within 2 seconds. A typical packet transmission time with Zigbee medium access control protocol and TOSSIM message payloads is about 2 ms [12]. Since SSA imposes forwarding within β seconds (or, approximately, within an average of $\beta/2$ seconds), we have

$$20 \cdot (2 \text{ ms} + \frac{\beta}{2} \text{ ms}) \leq 2000 \text{ ms} \quad \Rightarrow \quad \beta \leq 196 \text{ ms}.$$

We use $\beta = 0.196$ seconds in our analysis. Then,

$$\begin{aligned} p_d &= 1 - e^{-0.392/20} = 0.0194 \\ p &\geq 1 - e^{-0.196/20} = 0.0098, \quad \text{and} \\ p_{\text{real}} &= 0.0098/(0.0098 + 0.0194) \approx 1/3. \end{aligned}$$

Therefore, if a global adversary focuses only on the 2β-events, then one third of the corresponding transmissions are real packet transmissions. This significantly reduces the search space for the adversary.

Furthermore, the time limit of β seconds to inject or propagate a real packet results in significant time correlations of the transmissions by the nodes in the path taken by a real packet to reach a sink. On the other hand, if a dummy packet transmitted by a node, say n_i, triggers a 2β-event, the probability that n_i's neighbor will also transmit a dummy or real packet within β seconds of n_i's transmission is low, about 0.01.

We propose the following adversary strategy that combines these observations to detect sources.

Statistical Adversary (SA): The statistical adversary (SA) is a global adversary who records all timestamps of all transmissions and checks their distribution properties using Anderson-Darling or Kolmogorov-Simirnov tests to detect nodes that deviate from exponential distribution pattern for inter-message delays as described in [2]. SA supplements this with 2β-event detection logic. When a 2β-event is found, SA further analyzes the previous and future message transmission timestamps and geographical locations to construct a β-chain. A β-chain is a sequence of nodes $n_1, ..., n_k$ that have transmissions at times $t_1, t_2, ..., t_k$ satisfying the following properties: (a) the node, say n_j, that caused the 2β-event, at time t_j, is part of the chain; (b) n_i and n_{i+1}, $2 \leq i \leq k$, are within each other's radio range; and (c) $t_i - t_{i-1} \leq \beta + \delta$, where $2 \leq i \leq k$ and δ is a small value about 2-10 ms to account for delays induced by medium access control

```
event Heard(Node x) {
1.  TS[x][0] = TS[x][1]
2.  TS[x][1] = time()
3.  foreach node n_i ∈ NODE{
4.    if(n_i is a downstream neighbor of n_x
5.      && n_i in a suspected chain l
6.      && TS[x][1]-TS[i][1]≤ β + δ){
7.      Append n_x to l
8.      found = TRUE
9.    }
10. }
11. if(found == FALSE &&TS[x][1]-TS[x][0]≤2β)
12.   srcLocate(x, 1)    // See Figure 4
}
```

Fig. 3. Pseudocode to detect 2β-event and construct β-chain involving node n_x. TS[x] keeps the time stamps of the most recent two transmission of n_x. NODE is the set of all nodes in the sensor network. L is the set of all suspected chains in the sensor network and l is a member of this set. SA determines the ID of the node that transmitted a message and calls this function. If n_x is not part of a β-chain currently being constructed and transmitted within β seconds of a current β-chain head, the this node is appended to the list to form the new head. Otherwise, the timestamps of n_x are checked and, if applicable, a 2β-event is triggered and srcLocate() is called to construct a new β-chain from the source to n_x.

protocol and channel contention. SA treats each β-chain of 3 or more nodes as the route of real packet propagation to the sink with n_1 as the source and the sink within the radio range of n_k.

The pseudocodes to identify a 2β-event and construct the corresponding β-chain by SA are given in Figures 3 and 4.

We conclude this section with an estimation of $\overline{CT}$. Recall that we used

$$p \geq 1 - e^{-\lambda_e \beta},$$

see (2), to denote the probability that the occurrence of a real event or reception of a real message results in an observable 2β-event.

Let p_1 be the probability that a real 2β-event occurs at least once in the propagation of an event to sink. Then, $p_1 = 1 - (1-p)^h$.

The observation of a real event occurrence as a 2β-event may be modeled as a Bernoulli trial with p_1 as the probability of success. Then the number of expected real events that must occur before the detection of a 2β-event is the expectation of a modified geometric random variable with parameters p_1 and is given by $1/p_1 - 1$.

Then the expected capture time is

$$\overline{CT} = (\frac{1}{p_1} - 1) \cdot \frac{1}{\lambda_e}.$$

If we assume an average of $h = 10$ hops for the route used by a real event to reach sink (the scenario we simulated and discussed in Section V), then $p = 0.0098$, $p_1 = 0.094$, and $\overline{CT} = 193$ seconds.

```
int srcLocate(int x, int timepos){
1.  int i, j, ret, neighbor = 0, pos = 1
2.  float diff = 0, minTime = β + δ
3.  foreach node i ∈ NODE{
4.    if(i is a downstream neighbor of x){
5.      for(j = 0; j ≤ 1;j++){
6.        diff = TS[x][timepos]-TS[i][j]
7.        if(diff < minTime){
8.          neighbor = i
9.          pos = j
10.           minTime = diff
11.       }
12.     }
13.   }
14. }
15. if(neighbor ≠ 0) {
16.       ret = srcLocate(neighbor, pos)
17.       Append n_x to L[ret]
18.       return ret x
19. }else{ // found the β-chain source
20.   Initialize L[x] with n_x
21.   return x
22. }
}
```

Fig. 4. Construction of β-chain from the source to the node that triggered 2β-event. The transmissions heard until now are used to construct this part of the β-chain. The foreach loop identifies the most likely predecessor of the current node that must have sent a real packet. It recursively goes to the last possible node, and constructs the β-chain l, which will be a member of the set L[], during the return from recursive calls in the if-then part of the code.

V. Simulation results

We used the TOSSIM simulator to verify the capture time predictions given by the analytical model in Section IV and to gain additional understanding of SSA and SA. TOSSIM [17] simulates Crossbow MicaZ [18] like motes with TinyOS 2.0.1 [19].

We used a 12×12 grid of 144 sensor nodes with one sink at the top left corner of the grid. The nodes are numbered 1, ..., 12 for the first row, 13, ..., 24, for the second row, and so on. Node 1 is the sink. To reduce the occurrence of real 2β-events, we used a single source node—44 in one set of simulations and 101 in another set. (The simulation scenarios described later in this section show that having multiple real sources does not affect the detection capability by the adversary.)

The real events arrive at the source node at the rate $\lambda_e = \frac{1}{20}$ per second. All nodes transmit dummy packets with exponentially distributed IMDs with a mean of $1/\lambda_d = 20$. The simulation runs until 500 real packets are received by the sink. TOSSIM is modified to print a single line of timestamp and other information for each (dummy as well as real) transmission during the simulation.

A program based on the pseudocodes given in Figures 3 and 4 and mimics SA's timing analysis is fed with the TOSSIM output. The SA program reads each line only once and prints detected 2β-events and the corresponding β-chains. The number of 2β-events—real and dummy, β-chains of length 3 or more, the false positive rate—fraction of the instances false sources are identified, and the source detection rate—fraction of the instances the source is correctly identified are given in Figure 5.

A total of 1000 real events are injected in the two simulations with two different sources; 117 instances of them are correctly identified and the corresponding sources are located. So $\overline{CT} = \frac{1000/\lambda_e}{117} = \frac{1000*20}{117} = 171$ seconds. In Section IV, we estimated the $\overline{CT}$ for this case to be 193 seconds. We believe that the difference may be attributed to the approximation we used to calculate p.

Enhancing SSA

Based on our capture time analysis, it is clear that any solution that attempts to directly increase CT require the increase of the time-limit to trigger real packet forwarding by intermediate nodes. Such a modification must consider the event-deadlines imposed by the application supported by WSN. However, independent of the application requirements, a simple modification to SSA can be used to significantly increase CAs and the false positive rates. This in turn makes the adversary, who would like to remain undetected for long periods, cautious and effectively increases the capture time.

Modification: We require each node that injects a dummy packet marks it with some probability p_m as "must forward". These dummy packets will be propagated to the sink just like real packets with β seconds time limit for retransmissions by intermediate nodes.

The modification significantly increases the false positive rate. To verify this, we reran the simulations with the modification to SSA and analyzed the output with the program that implements SA's timing analysis. SA identified nearly all instances of injection of real packets and dummy packets marked for must-forward. (This shows that multiple true sources can be accurately identified by SA.) However, checking all such nodes significantly increases the false positive rates. The results presented in Figure 6 indicate that the false positive rate is 86% when 5% of dummy packets are marked as must-forward; the number of CAs will also increase from 53 to 430, more than 8 times. The false positive rates and the number of CAs are significantly higher with higher dummy packet marking probabilities.

VI. Conclusions

Communication privacy protocols that effectively hide the timing and locations of occurrences of real events are needed to protect WSNs from resourceful adversaries who can overhear transmissions in the entire network and employ detailed timing analysis techniques to identify the source nodes and sinks. In this paper, we investigated the effectiveness of two typical communication privacy protocols proposed recently that attempts to hide sources from either a local or a global adversary. Phantom routing, as an example against local adversaries, randomly routes packets to phantom sources and all packets follow shortest path from phantom sources, thus increasing the capture time. On the other hand, SSA, as an example against global adversary, makes nodes send dummy transmissions which can camouflage an occasional real packet transmissions.

We presented a detailed probabilistic model to analyze the expected capture time against both LAs and GAs. Our analytical model lead to a highly effective analysis strategy by an adversary.

Finally, we presented a simple but very effective improvement to SSA to increase the false positive rates experienced by the adversary by more than 8-fold. In future, we intend to develop more communication privacy protocols for more realistic scenarios in which sensors have low duty cycles and sleep most of the time.

References

[1] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *25th IEEE International Conference on Distributed Computing Systems, 2005. ICDCS 2005. Proceedings*, pp. 599–608, 2005.

[2] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *IEEE INFOCOM*, pp. 51–55, Citeseer, 2008.

[3] 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*, pp. 88–93, ACM, 2004.

[4] Y. Xi, L. Schwiebert, and W. Shi, "Preserving source location privacy in monitoring-based wireless sensor networks," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, p. 8, 2006.

Source	2β-events Detected			β-chains with 3+ nodes		β-chains starting at source	False Positive Rate	Source Detection Rate
	Total	Dummy	Real	Dummy	Real			
44	822	769	53	0	53	53	0/53 =0.0	53/53 = 1.0
101	794	729	65	0	65	64	1/65 = 0.015	64/65 = 0.985

Fig. 5. Detected dummy and real 2β-events and β-chains by SA. The adversary constructs a β-chain for each detected 2β-event, and checks only the nodes identified as sources of β-chains with 3 or more nodes. False positive rate is the fraction of the sources falsely identified as sources. True positive rate is the fraction of instances the source is correctly identified. The data from two simulations each with a different source node are indicated.

Marking Prob.	2β-events Detected			β-chains with 3+ nodes		β-chains starting at source	False Positive Rate	Source Detection Rate
	Total	Dummy	Real	Dummy	Real			
0.05	1022	961	61	369	61	61	369/430=0.858	61/430 = 0.122
0.10	1722	1662	60	1175	60	58	1177/1235=0.953	58/1235=0.047

Fig. 6. The modified privacy protocol significantly increases the false positive rates by SA. The source node is 44. The first column indicates the probability with which a dummy message is marked by its source as must-forward.

[5] J. Deng, R. Han, and S. Mishra, "Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks," *Pervasive and Mobile Computing*, vol. 2, no. 2, pp. 159–186, 2006.

[6] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *IEEE International Conference on Network Protocols, 2007. ICNP 2007*, pp. 314–323, 2007.

[7] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "Protecting receiver-location privacy in wireless sensor networks," in *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, pp. 1955–1963, 2007.

[8] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic," in *sensor networks, The ACM Conference on Wireless Network Security (WiSec*, Citeseer, 2008.

[9] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

[10] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of ACM SenSys*, 2003.

[11] H. Dai and R. Han, "TSync: a lightweight bidirectional time synchronization service for wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 1, p. 139, 2004.

[12] R. Boppana. and P. Pan, "A comparison of secure data aggregation schemes for wireless sensor networks," in *High Performance Computing (HiPC), 2009 International Conference on*, pp. 179 –188, dec. 2009.

[13] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, ACM New York, NY, USA, 2002.

[14] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pp. 29–42, ACM New York, NY, USA, 2004.

[15] O. Ugus, D. Westhoff, R. Laue, A. Shoufan, and S. Huss, "Optimized implementation of elliptic curve based additive homomorphic encryption for wireless sensor networks," in *2nd Workshop on Embedded Systems Security, WESS*, vol. 2007, 2007.

[16] H. Wang, B. Sheng, and Q. Li, "Privacy-aware routing in sensor networks," *Computer Networks*, vol. 53, no. 9, pp. 1512–1529, 2009.

[17] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.

[18] Crossbow Technology, Inc., "MICAz wireless measurement system." Document Part Number: 6020-0060-04 Rev A, http://www.xbow.com. Retrieved on May 2009.

[19] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler,

"TinyOS: An operating system for wireless sensor networks," *Ambient Intelligence*, 2004.