Int J Software Informatics, Volume 8, Issue 2 (2014), pp. 145–166
International Journal of Software and Informatics, ISSN 1673-7288
©2014 by ISCAS. All rights reserved.

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661040

Enhanced Source Location Privacy Based on Random Perturbations for Wireless Sensor Networks

Uthaiwan Srimongkolpitak[1], Yi Yang[2], and Hang Liu[1]

[1] (Department of Electrical Engineering and Computer Science,
The Catholic University of America, USA)

[2] (Department of Mathematics and Computer Science, Fontbonne University, USA)

Abstract Source location privacy, which means to protect source sensors' locations from being leaked out of observed network traffic, is an emerging research topic in wireless sensor networks, because it cannot be fully addressed by traditional cryptographic mechanisms, such as encryption and authentication. Current source location privacy schemes, assuming either a local or global attack model, have limitations. For example, the schemes under a global attack model are subject to a so called '01' attack, during which an attacker can potentially identify the sources of real messages. Targeting on tackling this attack, we propose two perturbation schemes, one based on Uniform Distribution and the other based on Gaussian Distribution. We analyze the security properties of these two schemes. We also simulate and compare them with previous schemes, with results showing that the proposed perturbation schemes can improve sensor source location privacy significantly. Furthermore, it is realized that an attacker may employ more intelligent statistical tools, such as Univariate Distribution based Reconstruction (UDR), to analyze the traffic generation patterns and find out real sources. We propose a Risk Region (RR) based technique, to prevent the attacker from successfully doing this. Performance evaluation shows that the RR-based scheme increases the errors of the attacker, so that the attacker is not able to accurately derive real messages as well as their sources.

Key words: source location privacy; wireless sensor networks; random perturbations; uniform distribution; Gaussian distribution; univariate distribution based reconstruction (UDR); risk region (RR)

Srimongkolpitak U, Yang Y, Liu H. Enhanced source location privacy based on random perturbations for wireless sensor networks. *Int J Software Informatics*, **Vol.8, No.2 (2014): 145–166. http://www.ijsi.org/1673-7288/8/i188.htm**

1 Introduction

Wireless Sensor Networks (WSNs) are being increasingly used in many important military and civilian areas from battlefield surveillance, to environmental monitoring, then to personal health maintenance, and so on. As WSNs become more pervasive, security and privacy requirements of many applications pose

This work is sponsored by the National Natural Science Foundation of China (No. 61370108).
Corresponding author: Yi Yang, Email: yyang@fontbonne.edu
Received 2014-06-10; Revised 2014-08-05; Accepted 2014-08-13.

additional challenges. WSNs in an unattended and even hostile environment are susceptible to many attacks[1].

Traditional research in WSNs security focuses on solving research issues, such as key establishment and management[3-8], data encryption/decryption, and message integrity/source authentication[9]. Most of these issues could be solved by using cryptographic techniques.

Recently, privacy preservation, which cannot be fully addressed by cryptographic techniques, such as encryption and authentication, has drawn a lot of researchers' attentions[10-19]. Privacy preservation includes protection of sources and receiver's location privacy. Since the receiver, i.e., the base station, is normally protected by tamper-proof mechanisms due to its importance, our focus in this paper is source location privacy, which means to protect the source nodes' locations.

Note that when messages are transmitted from sources to the destination (i.e., the base station), no matter how strong the cryptographic keys and encryption algorithms are, an observer can find out the locations of the message sources by performing traffic analysis. As an example shown in Fig. 1, in an asset monitoring network, when sensors detect an animal, they send messages reporting this event to the base station continuously. An adversary (i.e., a hunter here) can trace back to the sources hop by hop. He can then capture and even kill the animal. Similarly, in a battlefield scenario, the communications between soldiers and their surrounding sensors could reveal the positions of the soldiers, putting them in great danger.

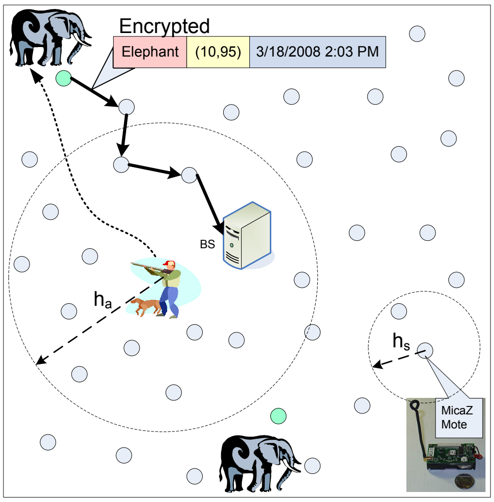

Figure 1. An application of sensor networks for animal monitoring, in which h_a is the hearing range of the attacker and h_s is the hearing range of regular sensors.

It is challenging to protect source location privacy in wireless sensor networks. Sensors operate under limited computational power, scarce processing capability, and little storage. Lightweight energy-efficient mechanisms are required. Furthermore, sensors communicate with each other through simple radio devices and normally in an open/broadcast manner, so it is easy for the attacker to overhear message transmissions among sensors. Most source privacy protection techniques designed for general networks[2] are not appropriate to be used for wireless sensor networks because either these techniques are too expensive to be deployed or the privacy issues in sensor networks are different.

Several solutions for preserving source location privacy in wireless sensor networks have recently been proposed in the literature[13,16]. However, these solutions have their limitations. Most work assumed a local attack model under which the attacker has a limited hearing range, comparable to that of regular sensors. Only a single transmitter is under the attacker's consideration at a time and the attacker tries to trace back to the source in a hop-by-hop fashion.

In practice, it is possible for an attacker to monitor all the network traffic either by deploying his own monitoring sensor network over the interested area or by employing a powerful site surveillance device with large enough coverage. The schemes developed for local attacks, such as phantom routing[13], perform poorly if the attacker becomes more powerful. For example, when the attacker's hearing range is more than three times of sensors' transmission range, the asset capture likelihood becomes as high as 97%[13].

If the attacker is able to monitor and analyze the traffic over the whole network, different techniques need to be developed to preserve the source location privacy. This problem has received relatively less research attention. Under such a strong global attack model, it is unlikely to achieve source location privacy if the traffic in the network are only real event messages. Sensors thus need to send encrypted dummy messages with the same size and format to hide real message transmissions. By observing message formats, the attacker cannot differentiate real and dummy messages. A straightforward method is that all the sensors periodically send dummy messages at a constant rate. When there is a real event, the source node postpones the real message transmission to the next time interval. However, in practice, it is difficult to determine an appropriate message transmission rate. If the rate is too high, there will be much overhead introduced by dummy messages; otherwise, if the rate is too low, there will be high latency for real message transmissions. Therefore, there is an inherent tradeoff among privacy, overhead, and latency here.

FitProbRate[16] is a state-of-the-art algorithm designed to preserve source location privacy under a global attack model. In FitProbRate, every node in the network sends out dummy messages with intervals following an exponential probability distribution that provides flexibility to reduce real message latency. When a sensor detects a real event, it sends the real event messages with intervals smaller but still following the same exponential distribution. This approach reduces the real event report latency while keeping the probabilistic distribution of message transmission intervals stable. However, if there are continuous real messages and/or real message rate is high, then the mean of message intervals for real message transmissions tends to be smaller than that of dummy messages.

To maintain the same exponential distribution of overall message intervals, in FitProbRate, transmissions of real event messages are normally followed by a recovery window in which dummy messages will be sent out with relatively larger intervals to recover the mean. Therefore, FitProbRate scheme suffers from a so called '01' test[20] although it prevents an attacker from using the change of the message transmission interval distribution to find out the locations of real event sources. If overtime an attacker accumulates an accurate population mean of all the observed message time intervals, he can identify all the message time intervals smaller than this mean (denoted as ‘0’s) and all the message time intervals larger than this mean (denoted as ‘1’s). Furthermore, he may find out that message time intervals smaller than this mean have a higher chance of coming from real sources. An intuitive way for the attacker to identify real sources is to identify all the ‘01’ patterns of message time intervals and to derive that ‘0’s in these patterns are likely from real messages/sources.

In this paper, we propose schemes to use perturbations on message transmission time intervals to protect source location privacy which solves the ‘01’ attack problem in existing schemes. More specifically, we propose two perturbation schemes, one based on Uniform Distribution and the other based on Gaussian Distribution. By adding perturbations to the message transmission intervals, the message intervals observed by the attacker randomly deviates from the population mean, which makes the ‘01’ patterns obscure and lowers the attacker’s detection capability. We analyze the security properties of these two schemes. We also compare their performance with that of the existing scheme, and our results show that the proposed perturbation schemes can improve the source location privacy significantly. Furthermore, it is realized that an attacker may employ more intelligent statistical tools, for example, Univariate Distribution based Reconstruction (UDR)[29], to analyze the traffic generation patterns, reconstruct the original message intervals, and thus find out real sources. We propose a Risk Region (RR)-based solution to prevent the attacker from accurately/successfully doing this.

The rest of the paper is organized as follows. We describe the system model in Section 2. We propose two perturbation schemes and analyze their security characteristics in Section 3. The UDR technique that the attacker may employ and the Risk Region (RR)-based scheme are described in Section 4. Then, we evaluate the two perturbation schemes and the RR-based scheme, and compare their security performance with the state-of-the-art scheme, FitProbRate, in Section 5. Related work is discussed in Section 6. Conclusion and future work are given in Section 7.

2 System Models

We first introduce our network model as well as adversary model.

2.1 Network model

We consider that n sensor nodes are randomly distributed in the deployment area. There are $n'(0 < n' < n)$ out of n nodes detecting real events and sending real messages simultaneously. Once a real event is detected, real messages containing real event related information, such as event type, location, and time will be sent from sources to the base station. Dummy messages are generated to cover these real messages. All messages are encrypted and of the same format. Base station resides in a fixed location of the network, e.g., at the center.

2.2 Attack model

Since all the messages are encrypted/scrambled and appear random to the attacker, the attacker has to try to identify real messages from their time intervals because real sources tend to send out real messages as soon as possible to reduce the latency. In our attack model, the attacker has a hearing range h_a, which is multiple times larger than that of the regular sensors h_s (i.e., $3h_s < h_a \leqslant r$, where r is network radius), e.g., a laptop-class attacking device with more powerful but limited hearing capabilities. We assume that maximally the attacker can observe and analyze w message time intervals altogether, i.e., the attacker's window size is $w \geqslant 0$. We differentiate sensor's transmission range (t_s) and hearing range (h_s), i.e., t_s might not equal to h_s. Like other papers in the same area[14-16], we assume that the base station cannot be compromised.

The attacker might compromise a (small) fraction of sensor nodes to stop their normal operations and obtain their security credentials. Although the attacker is able to decrypt messages from these compromised sensors, these compromised sensors might not necessarily be real sources. So, by fully controlling a small number of compromised sensors the attacker can still do very limited things in the network.

3 Perturbation Schemes

As introduced formerly, the previous schemes under a global attack model are subject to a so called '01' attack. Focused on defending against this attack, we present two source location privacy schemes based on random perturbations: one based on Uniform Distribution and the other based on Gaussian Distribution.

These two schemes work because based on random perturbations the sample mean calculated by the attacker becomes inaccurate although the population mean will not change, which means sample means deviate from the true or population mean. In this way, the attacker cannot accurately identify all the '01' patterns, which decreases the attacker's detection capability for real message time intervals significantly.

3.1 Uniform distribution based perturbation scheme

Similar to Ref. [16], the baseline of message time intervals follows an Exponential Distribution, because according to its probability density function a traffic generator following an Exponential Distribution tends to generate small time intervals. Our basic idea is to add random perturbations to each individual time interval that follows an Exponential Distribution. As shown in Fig. 2, every time interval will follow the probabilistic distribution of $X + R$ instead of only X, where X is the Exponential Distribution and R is the perturbation distribution.

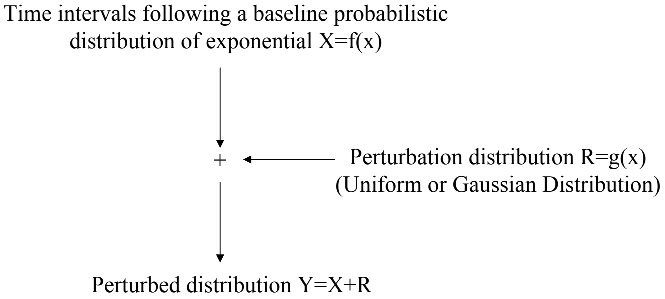
Time intervals following a baseline probabilistic distribution of exponential X=f(x)
+
Perturbation distribution R=g(x) (Uniform or Gaussian Distribution)
Perturbed distribution Y=X+R

Figure 2. Adding perturbations into individual message time intervals.

The common perturbations normally follow two different distributions. The first one is a Uniform Distribution with 0 as mean and range is $[-a, +a]$. The second perturbation could be a Gaussian Distribution, which will be discussed in the next section (Section 3.2).

Every time a real message comes, a small interval with maximal negative perturbation, e.g., $-a$ in Uniform Distribution, could be assigned. Note that we still need to guarantee time intervals from real messages are positive and small. The technique to implement this perturbation is presented in Algorithm 1.

Algorithm 1 Perturbations based on uniform distribution

Input: uniform distribution in range $[-a, a]$; rate r of real event; duration d of real event; mean μ of the exponential distribution;

Output: m perturbed message time intervals $\lambda_i (0 \leqslant i \leqslant m-1)$ including time intervals of real event messages;

Procedure:

```
val = round($1/r$); {function *round* changes the result of $1/r$ to the nearest integer}
for $i = 0$ to $m - 1$ do
  if rem($i$, val)=0 then
    for $j = 0$ to $d - 1$ do
      $\lambda_{ij}$ = exprnd($\mu$)-$a$; {maximally negative perturbations for real messages;
        function *exprnd* generates random numbers following an exponential
        distribution with mean $\mu$}
      while $\lambda_{ij}$ <= 0 do
        $\lambda_{ij}$ = exprnd($\mu$)-$a$; {time intervals should be positive}
      end while
    end for
  else
    $\lambda_i$ = exprnd($\mu$)+unifrnd((-1)×$a$, $a$); {regular perturbations following a
      uniform distribution; function *unifrnd* generates random numbers following
      a uniform distribution in the range of $(-a, a)$}
    while $\lambda_i$ <= 0 do
      $\lambda_i$ = exprnd($\mu$)+unifrnd((-1)×$a$, $a$); {time intervals should be positive}
    end while
  end if
end for
```

3.2 Gaussian distribution based perturbation scheme

The second perturbation could be a Gaussian Distribution with 0 as mean and standard deviation σ. Every time a real message comes, a small interval with maximal negative perturbation, e.g., $-\sigma$, $-2 \times \sigma$, or even $-3 \times \sigma$ in Gaussian Distribution, could be assigned. Once again, we still need to guarantee that the time intervals from real messages are positive and small. The details of implementation are presented in Algorithm 2.

In order to guarantee that time intervals are positive, the basic idea of our algorithm is that every time we generate a random time interval, we check whether it is positive or not; if not, we repeat this process until the random time interval is positive. Since the operations of random number generator are simple, we can find positive random values very fast.

Algorithm 2 Perturbations based on gaussian distribution

Input: standard deviations σ; rate r of real event; duration d of real event; mean μ of the exponential distribution;

Output: m perturbed time intervals $\lambda_i (0 \leqslant i \leqslant m-1)$ including time intervals of real event messages;

Procedure:

```
val = round($1/r$);
for $i = 0$ to $m-1$ do
  if rem($i$, val)=0 then
    for $j = 0$ to $d-1$ do
      $\lambda_{ij}$ = exprnd($\mu$)+$b$; {maximally negative perturbation for real messages when $b$ is $-\sigma, -2\times\sigma$, or $-3\times\sigma$}
      while $\lambda_{ij} <= 0$ do
        $\lambda_{ij}$ = exprnd($\mu$)+$b$; {time intervals should be positive when $b$ is $-\sigma, -2\times\sigma$, or $-3\times\sigma$}
      end while
    end for
  else
    $\lambda_i$ = exprnd($\mu$)+normrnd(0, $\sigma$); {regular perturbations following a gaussian distribution; function normrnd generates random numbers following a gaussian distribution with mean 0 and standard deviation $\sigma$}
    while $\lambda_i <= 0$ do
      $\lambda_i$ = exprnd($\mu$)+normrnd(0, $\sigma$); {time intervals should be positive}
    end while
  end if
end for
```

3.3 Security evaluation

Let us take a look at the probability for ‘01’ patterns appearing in FitProbRate scheme[16]. Suppose m is the total number of messages and r is real message rate. There are two situations for ‘01’ patterns to occur: if there are real events, then the probability for ‘01’ pattern to appear is 1; otherwise, the probability for ‘01’ pattern to appear is 1/4 because the probability for two continuous messages to be the pattern of ‘00’, ‘01’, ‘10’, or ‘11’ is equal. Hence, according to Total Probability Formula and Classical Probability Model, the probability $p(x_1)$ for ‘01’ patterns to appear in FitProbRate scheme is as follows:

$$\begin{aligned} p(x_1) &= \frac{m \times r + (m - m \times r) \times \frac{1}{4}}{m-1} \\ &= \frac{3 \times m \times r + m}{4 \times (m-1)} \\ &\text{when m is large} \\ &\approx \frac{3}{4} \times r + \frac{1}{4}. \end{aligned} \tag{1}$$

Below, we analyze the probability for '01' patterns to appear in our schemes. The probability $p(x_2)$ for '01' pattern to appear is 1/4 because due to our random perturbations, the probability for two continuous messages to be the pattern of '00', '01', '10', or '11' is equal, i.e.,

$$p(x_2) = \frac{1}{4} < p(x_1), \tag{2}$$

since $r > 0$. This means that the probability for '01' patterns to occur in our schemes is smaller than that in the FitProbRate scheme and the actual difference depends on the real message rate.

We will use simulations in Section 5.3.1 to validate our analytical results. The probability for '01' patterns to appear in our perturbation schemes is close to 1/4, which is the probability for any random patterns of two continuous message intervals to appear. This means that by perturbations the '01' patterns of real message time intervals are hidden well in a large quantity of dummy message time intervals. Also, this probability is smaller than that in the FitProbRate scheme, which means that our perturbation schemes can reduce the probability of '01' patterns to occur from a relatively large value (though the difference depends on the real message rate) to a value for random patterns; so, the attacker cannot gain anything from purely identifying '01' patterns in our perturbation schemes.

4 Risk Region (RR) Based Scheme

As indicated before, the attacker might use some intelligent statistical tools such as Univariate Distribution based Reconstruction (UDR)[29] to reconstruct the original message time intervals before perturbation. In the following, we will discuss both this offensive side (Section 4.1), which we call it UDR attacking technique, and the defensive side (Sections 4.2), which we call it Risk Region or RR based scheme.

4.1 UDR-based attacking technique

First, we analyze what is known and unknown to the attacker. From the attacker's point of view, $Y = X + R$, where Y represents the perturbed message time intervals, X represents the original message time intervals, and R represents the perturbations. Y could be obtained directly from the sample message time intervals. R follows either Uniform or Gaussian Distribution with mean to be zero. However, the parameter for Uniform Distribution, i.e., range a is unknown to the attacker. Also, the other parameter for Gaussian Distribution, i.e., standard deviation σ is unknown to the attacker. X follows an Exponential Distribution, with the only parameter – mean μ_X known to the attacker, because $\mu_X = \mu_Y - \mu_R = \mu_Y - 0 = \mu_Y$, where the sample mean μ_Y could be directly obtained from sample perturbed message time intervals. Note: although the attacker knows the only parameter of the original message time intervals following an Exponential Distribution, the attacker cannot reconstruct every original message time intervals because the attacker does not know the seed to generate these original probabilistic message time intervals.

Then, the next question that the attacker might want to answer is: from sample or perturbed message time intervals Y and perturbations R, how to reconstruct original message time intervals X? An intuitive way to do this is by using formula $X = Y - R$. Although Y is known, the other parameter of perturbation R is unknown, so the attacker is not able to obtain each individual datum of perturbation R and each individual datum of distribution X cannot be reconstructed in this way either.

In literature, there is a statistical method, called Univariate Distribution-based Reconstruction (UDR)[29], applying here. The basic idea of UDR is as follows. The attacker can first derive the posterior distribution $P(X|Y)$, which gives him the probability for different values of X after having observed the value of Y. Since his goal is to reconstruct the original message time intervals, he needs to pick a value that can minimize the overall mean square error. There is a theorem in Ref. [29] stating that if he uses expected values $E(X|Y)$ to estimate or reconstruct the original message time intervals he will be able to minimize his mean square errors. Correlations might be able to further increase the accuracy of reconstruction. However, all the probabilistic distributions, such as X and R, are independent, so correlations do not apply in our case.

The attacker's mean square errors that we consider in this paper are defined as following:

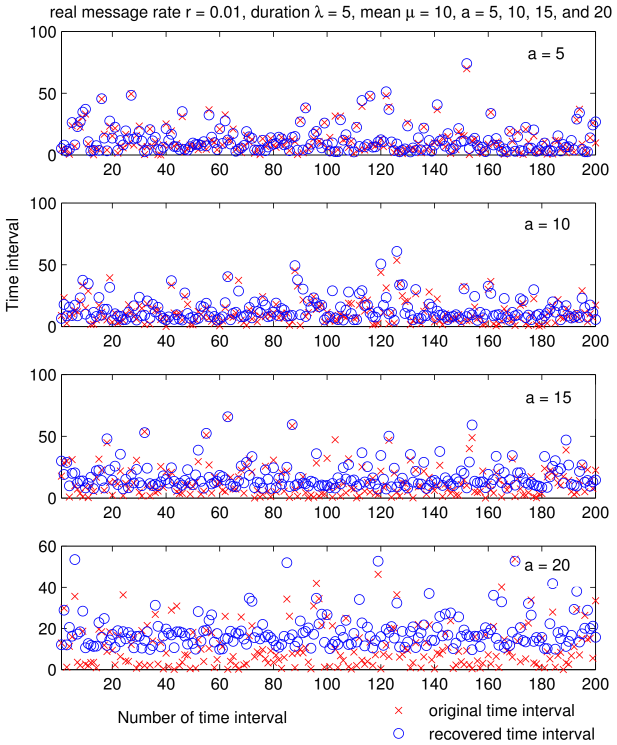

Figure 3. Comparison between reconstructed original time intervals and actual time intervals for Uniform Distribution-based perturbation scheme in terms of multiple a values.

Definition 1. Let x'_i and x_i be the attacker's recovered and original message time intervals respectively, where $i = 1, \dots, \delta$ and δ is the sample size. Then, the mean square errors (m.s.e.) of the attacker could be defined as:

$$m.s.e. = \frac{1}{\delta}\sum_{i=1}^{\delta}(x_i' - x_i)^2.$$

The comparison between reconstructed original message time intervals and actual original message time intervals for Uniform Distribution-based perturbation scheme is presented in Fig. 3. The comparison between reconstructed original message time intervals and actual original message time intervals for Gaussian Distribution-based perturbation scheme is in Fig. 4. From Figs. 3 and 4, we can see that by using UDR the attacker is able to reconstruct the original message time intervals in a relatively accurate way especially when the perturbation parameters are small, i.e., when a for Uniform Distribution and σ for Gaussian Distribution are small. This motivates our proposed Risk Region based scheme in the next section (Section 4.2).

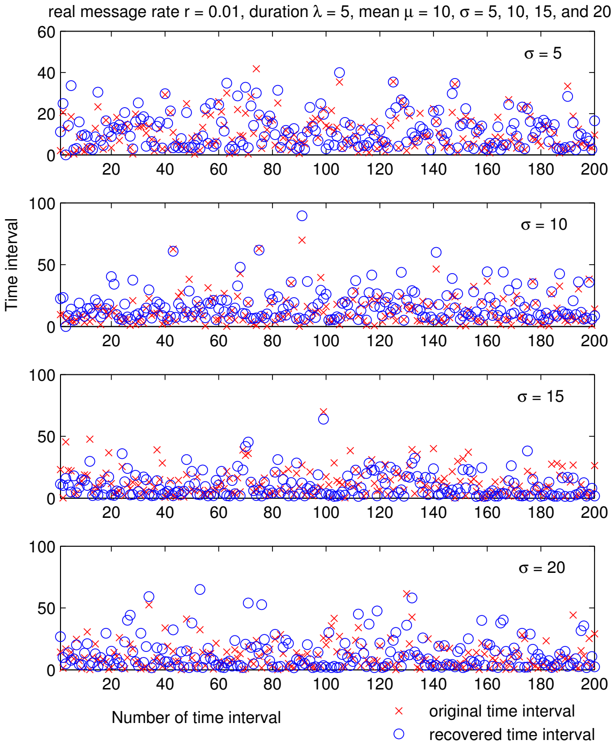

Figure 4. Comparison between reconstructed original time intervals and actual time intervals for Gaussian Distribution-based perturbation scheme in terms of multiple σ values.

4.2 Risk region (RR) algorithm

We propose a Risk Region (RR)-based scheme to defend against attacker's UDR attacking technique. The basic idea of RR is to increase the variance of our original message time intervals (i.e., data values of X), so that the attacker's mean square errors will be increased accordingly. According to the definition of mean square errors, it is not hard for us to understand that if we can somehow increase original message time intervals X's variance the attacker's mean square errors could be increased too, because attacker's mean square errors directly relate to X. The idea of RR is presented in Algorithm 3.

Algorithm 3 Risk Region Algorithm

Input: perturbation parameter β (β is either a in Uniform Distribution-based perturbation scheme or σ in Gaussian Distribution-based perturbation scheme); RR parameter p;

Output: qualified perturbed message time interval y;

Procedure:

```
threshold $b' = \beta \times p$;
calculate a $x$; {calculate an original message time interval following an Exponential Distribution}
calculate a $r$; {calculate a perturbation following an either Uniform or Gaussian Distribution}
$y = x + r$;
while $(|r| \leqslant b')||(y \leqslant 0)$ do
    calculate a new $x$; {we have two requirements for valid perturbed message time intervals: $|r| > b'$ and $y > 0$}
    calculate a new $r$;
    $y = x + r$;
end while
return $y$;
```

Basically, we set up a threshold b'. We accept the generated perturbed message time interval $y = x + r$ only if two requirements have been met: the absolute value of r is larger than this threshold b' and y is larger than 0. Otherwise, we will recalculate x and r until there is a valid y that has been generated. Based on our experiments, we choose the threshold b' to be the perturbation parameter β (β is a in Uniform Distribution or σ in Gaussian Distribution) *times* an RR parameter p and $p = 0, \frac{1}{4}, \frac{2}{4}$ and $\frac{3}{4}$.

Next, we explain why the RR algorithm can increase X's variance. There are five cases that we need to consider:

1. if $r = 0$, the first requirement $|r| > b'$ cannot be met, so x and r will be recalculated;

2. if $r > 0$ and $r <= b'$, the first requirement $|r| > b'$ cannot be met, so x and r will be recalculated;

3. if $r > 0$ and $r > b'$, the first requirement is met. Because both x and r are larger than 0, the second requirement is also met. Therefore, we will accept this y;

4. if $r < 0$ and $-r <= b'$, the first requirement $|r| > b'$ cannot be met, so x and r

will be recalculated;

5. if $r < 0$ and $-r > b'$, we will make $x > -r$ so that $y = x + r > 0$.

All the first four cases are either recalculation or acceptance. The key point is case 5, in which we increase the variance of X by enlarging x in a certain degree.

To apply the RR technique to the previous two perturbation schemes, we revise Algorithms 1 and 2 as follows:

Changes Made to Algorithm 1

```
11: $x_i = exprnd(\mu)$;
12: $r_i = unifrnd((-1) \times a, a)$;
13: $\lambda_i = x_i + r_i$;
14: while $(\lambda_i <= 0) || (|r_i| <= b')$ do {We add one more while condition here:
    $(|r_i| <= b')$}
15:   $x_i = exprnd(\mu)$;
16:   $r_i = unifrnd((-1) \times a, a)$;
17:   $\lambda_i = x_i + r_i$;
18: end while
```

In more detail, lines 11 to 16 in Algorithm 1 should be replaced with lines 11 to 18 in above “Changes Made to Algorithm 1”.

Changes Made to Algorithm 2

```
11: threshold $b' = \beta \times p$;
12: $x_i = exprnd(\mu)$;
13: $r_i = normrnd(0, \sigma)$;
14: $\lambda_i = x_i + r_i$;
15: while $(\lambda_i <= 0) || (|r_i| <= b')$ do {We add one more while condition here:
    $(|r_i| <= b')$}
16:   $x_i = exprnd(\mu)$;
17:   $r_i = normrnd(0, \sigma)$;
18:   $\lambda_i = x_i + r_i$;
19: end while
```

In more detail, lines 11 to 16 in Algorithm 2 should be replaced with lines 11 to 19 in above “Changes Made to Algorithm 2”.

We will evaluate the performance (especially the attacker’s mean square errors) before and after using the RR algorithm in Section 5.3.2.

5. Performance Evaluation

In this section, we first introduce the setup of our simulation and our evaluation metrics before we present our simulation results.

5.1 Simulation settings

In our simulation settings, there are 100 nodes randomly distributed in the deployment area. Out of them, there are 5 real sources detecting real events. By default, a real event lasts for 5 messages (i.e., duration $\lambda = 5$). We choose the rates of real event to be 0.01, 0.02, 0.04, 0.05, 0.1. The mean of the exponential distribution is 10. By default, the perturbation parameters (a) in Uniform distribution and (σ) in Gaussian distribution are both 5. For the attacker, his attacking window size is 1000 by default. The attacker uses one-sample Kolmogorov-Smirnov test as his goodness of fit test and the significance level (α) in his statistic test is 5%.

5.2 Simulation metrics

In our simulation, a detection is defined as "a '01' pattern has been identified". Then, the attacker's detection rate is formulated as follows:

Definition 2. Suppose the number of *detected* '01' patterns caused by real messages is denoted as t and the *actual total* number of '01' patterns caused by real messages in the traffic is denoted as $t'(t \leqslant t')$, then the detection rate of the attacker is defined as:

$$detection\ rate = t/t'.$$

Here, the attacker's false positive rate is formulated as following:

Definition 3. Suppose the total number of detected '01' patterns by the attacker is v, the total number of sensors is n, and the number of messages from each sensor is m, then the false positive rate of the attacker is defined as:

$$false\ positive\ rate = \frac{v - t}{m \times n - t'}.$$

To better understand the effectiveness of the attacker's detection, we check the Bayesian detection rate[21] of the attacker, which is defined as the probability for an alarm to really indicate a real message. In more detail, we have the following definition:

Definition 4. Suppose the total number of detected '01' patterns by the attacker is v and the number of detected '01' patterns caused by real messages is t, then the Bayesian detection rate of the attacker is defined as:

$$Bayesian\ detection\ rate = t/v.$$

5.3 Simulation results

This section is divided into two parts: the first will show the performance of perturbation schemes based on both Uniform Distribution and Gaussian Distribution (Section 5.3.1); the second will show the performance of RR-based scheme (Section 5.3.2).

5.3.1 Results on perturbation schemes

We first use simulation results to validate our security analysis. Then, we compare the performance of two perturbation based schemes and also compare them with the previous schemes. Last, we evaluate the impact of different parameters.

- **Validation of security analysis:** First, we run simulations to validate the results of our security analysis. In Fig. 5 and Fig. 6, we run the simulations in 100 trials to derive the probability of '01' patterns in our proposed schemes. From Fig. 5, we can see that the probability of '01' patterns for the Uniform Distribution scheme changes around 0.238, which is close to the theoretical result $\frac{1}{4}$. Similarly, the probability of '01' patterns for the Gaussian Distribution scheme changes around 0.237, which is also close to the theoretical result $\frac{1}{4}$, as shown in Fig. 6. These match the results of our security analysis well.

Also, from Fig. 7, we can see the probability of '01' patterns in percentage changes with the real message rates for the FitProbRate scheme. Again, the simulation results are close to the results of our security analysis, which validate our security analysis.

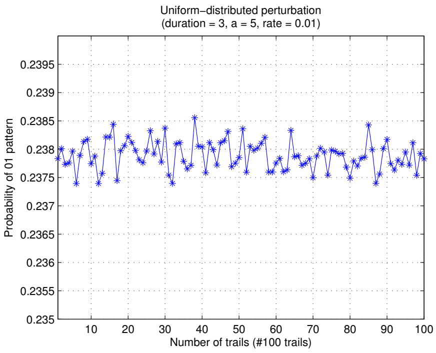

Figure 5. Validating analysis on Uniform distribution.

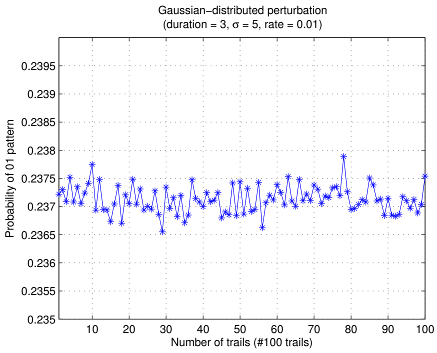

Figure 6. Validating analysis on Gaussian distribution.

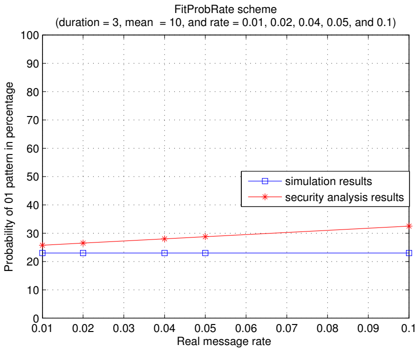

Figure 7. Validating analysis on FitProbRate scheme.

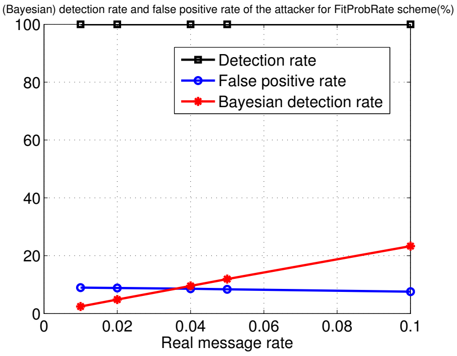

Figure 8. The FitProbRate scheme.

- **Comparison with previous schemes:** Overall, from Fig. 8, Fig. 9, and Fig. 10, we can see that random perturbations following either Uniform Distribution or Gaussian Distribution can decrease the attacker's detection rate and Bayesian detection rate, and also increase the attacker's false positive rate significantly.

For example, from Fig. 8 and Fig. 9, perturbations following Uniform Distribution can decrease the attacker's detection rate from around 100% to around 20%. They can also decrease the attacker's Bayesian detection rate from around 20% to less than 5%.

From Fig. 8 and Fig. 10, we can see that perturbations following Gaussian Distribution can further decrease the attacker's performance. For example, the attacker's detection rate is decreased to around 15% and the attacker's Bayesian detection rate is decreased to less than 5%.

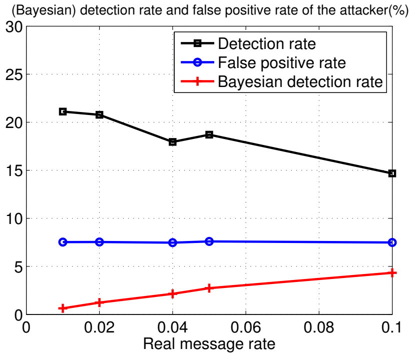

Figure 9. Uniform perturbations.

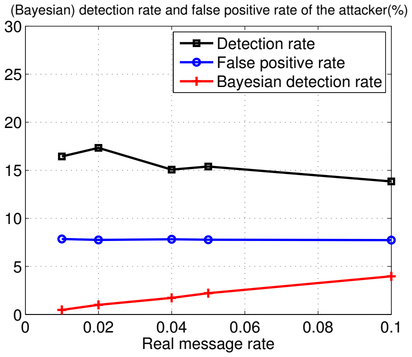

Figure 10. Gaussian perturbations.

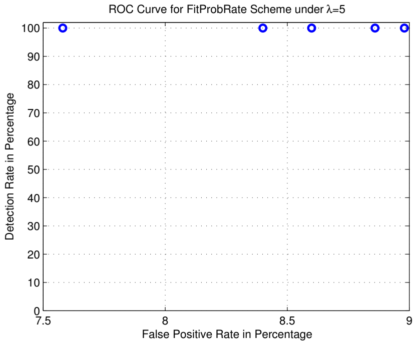

Figure 11. ROC for FitProbRate scheme.

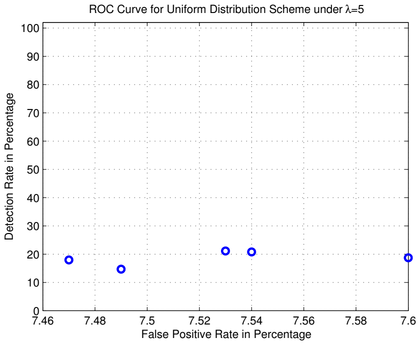

Figure 12. ROC for Uniform Distribution Scheme.

From Fig. 11, Fig. 12, and Fig. 13, we can see that the perturbations can make the attacker's detection much less effective, because perturbations significantly lower the attacker's Receiver Operating Characteristic (ROC) curve, which is a plot of the detection rate against the false positive rate. The attacker's detection rate decreases from around 100% to around 20% and around 15% by perturbations following Uniform Distribution and Gaussian Distribution, respectively.

- **Comparison of two perturbation schemes:** Comparing from the Uniform Distribution scheme to the Gaussian Distribution scheme, the attacker's detection rate and Bayesian detection rate are lower from around 20% to around 15% and the false positive rate is slightly higher because the scheme with perturbations following Gaussian Distribution has slightly better performance than the Uniform Distribution scheme. All of these can be seen from Fig. 9 to Fig. 10 and from Fig. 12 to Fig. 13.

- **Impact of different parameters:** From Fig. 8, we can see that in the FitProbRate scheme if the real message rate is higher then the attacker's performance is better: his detection rate and false positive rate remain almost the same, but his Bayesian detection rate is higher. Obviously, in this scheme, if real messages appear more frequently with a higher real message rate, this will make the real sources more easily to be detected.

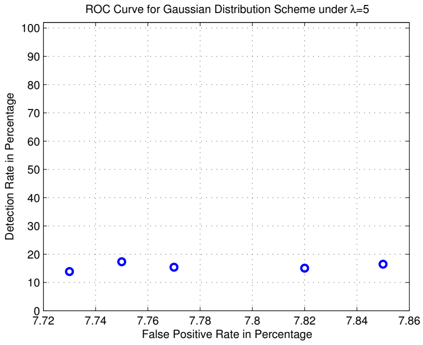

Figure13. ROC for Gaussian Distribution Scheme.

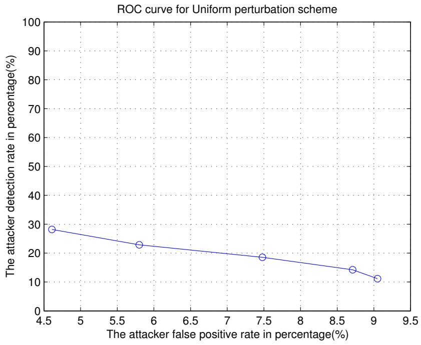

Figure 14. ROC curve for Uniform Perturbation scheme (impact of a).

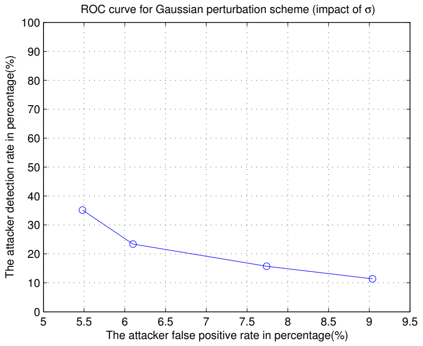

Figure 15. ROC curve for Gaussian Perturbation scheme (impact of σ).

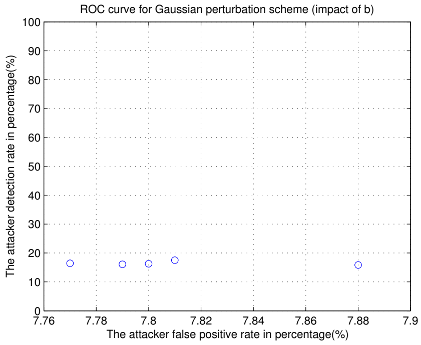

Figure 16. ROC curve for Gaussian Perturbation scheme (impact of b).

From Fig. 9 and Fig. 10, we can see the impact of real message rate on our perturbation schemes. In both schemes, when real message rate increases, the attacker's false positive rate almost remains the same, but his Bayesian detection rate increases at the cost of a decreasing detection rate. Therefore, the attacker cannot benefit from an increasing real message rate, which shows one advantage of our perturbation schemes.

From Fig. 14 and Fig. 17, we can see the impact of parameter a in Uniform Perturbation scheme. When a increases, the attacker has a better performance: he has a higher detection rate, a higher Bayesian detection rate, and a lower false positive rate. Therefore, when we choose the values for parameter a, it does not mean that a larger a is necessarily better.

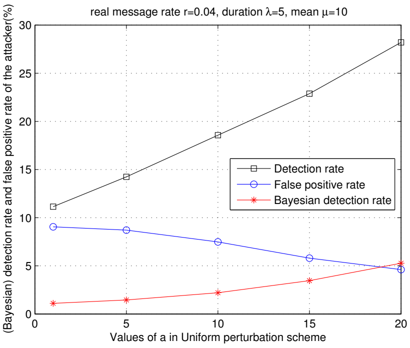

Figure 17. Impact of a in Uniform Perturbation scheme.

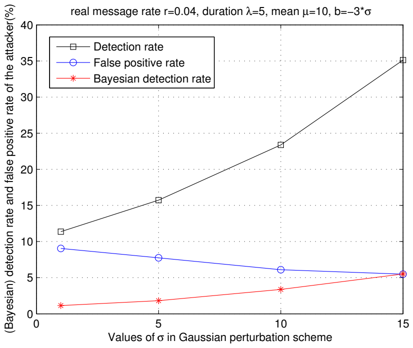

Figure 18. Impact of σ in Gaussian Perturbation scheme.

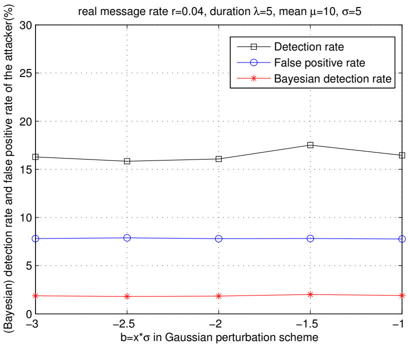

Figure 19. Impact of b in Gaussian Perturbation scheme.

We notice that if $a = 0$, our scheme becomes a perfect-privacy scheme, because both real and dummy messages follow the same exponential distribution. However, in this scheme, real message latency is high since real message time intervals are not perturbed. Our scheme makes real and dummy message time intervals almost equivalently small. Although it is not as secure as the pure exponential scheme, our scheme trades a certain degree of security for performance. On the other hand, if a is larger, there are larger differences between real and dummy message time intervals, which makes the '01' patterns more obvious. Hence, the attacker has better performance. Also, a larger a means longer running time for our algorithm because it is harder to find positively small time intervals for the real messages, so we should choose an appropriately small value for a (such as $a = 5$).

From Fig. 15 and Fig. 18, we can see the impact of parameter σ in Gaussian Perturbation scheme. It is similar to the impact of a in the Uniform Perturbation scheme. When σ is larger, the attacker has a better performance, so when we determine parameter values we should choose an appropriate (small) value (such as 5) for σ, to decrease the attacker's performance.

From Fig. 16 and Fig. 19, we can see the impact of parameter b in Gaussian Perturbation scheme. The attacker's performance does not change much with the values of b, compared with the other two parameters a and σ.

5.3.2 Results on risk region based scheme

We first check attacker's mean square errors under different RR parameter p values for both Uniform and Gaussian Distribution-based Perturbation Schemes. Then, we check attacker's mean square errors under different perturbation parameter a values for Uniform Distribution-based Perturbation Scheme and under different perturbation parameter σ values for Gaussian Distribution-based Perturbation Scheme, after RR-based technique is applied.

- **Attacker's performance before and after using RR:** The attacker's performance before using RR could be seen when $p = 0$, in which attacker's mean square errors are relatively small. From Fig. 20 and Fig. 21, we can see that after RR-based technique is used overall attacker's mean square errors increase with perturbation parameters (a in Uniform Distribution-based Perturbation Scheme and σ in Gaussian Distribution-based Perturbation Scheme). More important, when RR parameter p is larger, the attacker's mean square errors will also become larger. This shows that RR technique can effectively increase the attacker's errors when he reconstructs or recovers original message time intervals.

- **Attacker's performance under different perturbation parameters:** Figures 22 and 23 further validate that attacker's mean square errors increase with perturbation parameters: a for Uniform Distribution-based Perturbation Scheme and σ for Gaussian Distribution-based Perturbation Scheme, after RR-based technique has been applied. Therefore, in RR-based scheme, by increasing the scale of our perturbations, we can also increase the attacker's reconstruction errors for original message time intervals.

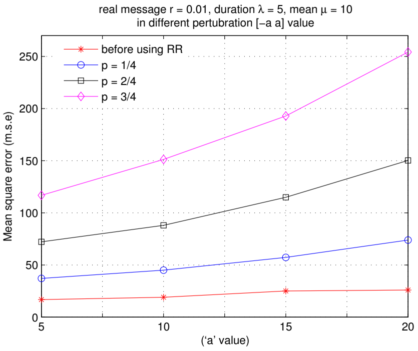

Figure 20. Attacker's mean square errors in Uniform Distribution-based Perturbation Scheme with different p values before and after RR-based technique has been applied.

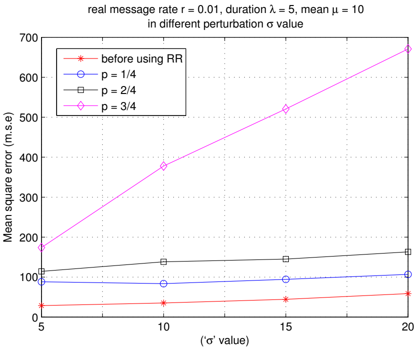

Figure 21. Attacker's mean square errors in Gaussian Distribution-based Perturbation Scheme with different p values before and after RR-based technique has been applied.

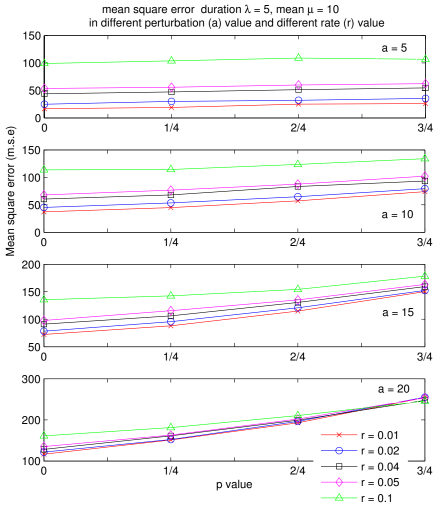

Figure 22. Attacker's mean square errors in Uniform Distribution-based Perturbation Scheme with different a values after RR-based technique has been applied.

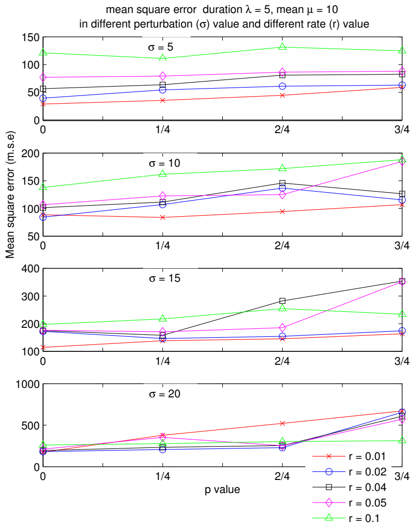

Figure 23. Attacker's mean square errors in Gaussian Distribution-based Perturbation Scheme with different σ values after RR-based technique has been applied.

6 Related Work

In general, privacy preservation in wireless sensor networks could be divided into protecting either sources' or receiver's locations. References [22,23,24,17] employ countermeasures against traffic analysis to improve receiver's location privacy, whereas our focus is source location privacy.

To improve source location privacy, Ref. [13] proposes phantom routing technique, in which messages are first forwarded by single-path random walk, then they are flooded in the area to reach the base station. Although by employing phantom routing the safety period is significantly improved, when the attacker's hearing range is increased to more than three times larger than that of regular sensors, the targets or victims' capture likelihood is increased to over 97% correspondingly.

References [25] and [26] consider a laptop-class eavesdropper in their attack model. Reference [25] proposes four schemes: naive, global, greedy, and probabilistic, to deal with laptop-class attacks. Periodic collection and source simulation are proposed in Ref. [26] to protect the context information under a global eavesdropper. Reference [18] considers node compromise attack and proposes a one-way hash chain based scheme to randomly select intermediate nodes transforming the packets, in order to obfuscate the transmission links from source to destination.

In Ref. [16], a global attack model is under consideration. To protect source privacy under such a strong attack model, extra message overhead (i.e., dummy messages) has to be introduced. Otherwise, if all the messages in the network are real messages, then the attacker will know that every message transmission signals a real event. Under this model, the simplest scheme has a constant rate. However, the difficulty in determining this rate reflects a tradeoff among message overhead, real event report latency and source location privacy. This paper proposes a FitProbRate scheme to reduce real event report latency by relaxing a certain degree of privacy.

Reference [20] also discusses the performance of FitProbRate scheme under 01 test or attack. However, its focus is to deal with the case of high-rate, continuous real messages. The proposed dynamic mean scheme has a better performance (i.e., the attacker has a higher false positive rate and a lower Bayesian detection rate) in this case. Our perturbation- and RR-based schemes are more general, compared with this scheme.

Besides these, Ref. [10] gives a state-of-the-art survey in privacy preservation techniques for wireless sensor networks. Reference [12] introduces buffering delay to provide temporal privacy, which is suitable for delay-tolerant applications of wireless sensor networks. Reference [27] proposes a cross-layer solution in which the event information is first propagated several hops through a MAC-layer beacon. Then, it is propagated at the routing layer to the destination to avoid further beacon delays. To improve source location privacy, Ref. [28] proposes dynamic routing schemes, in which messages are first transmitted to randomly selected intermediate nodes to confuse the attacker.

7 Conclusion and Future Work

Recently, source location privacy has become an important research topic for wireless sensor networks. Previous techniques used to protect source location privacy in wireless sensor networks were not adequately effective.

Focused on solving the problems found in previous schemes, we propose two perturbation schemes that can effectively decrease the attacker's detection capability on real event messages: one based on Uniform Distribution and the other based on Gaussian Distribution. Our simulation results show that our random perturbation schemes can improve source location privacy significantly compared with previous work, and the scheme based on Gaussian Distribution has slightly better security performance than the scheme based on Uniform Distribution, at the cost of more computational consumption.

Furthermore, an attacker can use some intelligent statistical tools, such as Univariate Distribution based Reconstruction (UDR), as an attacking or recovering technique, to identify real messages by reconstructing original message time intervals. Then, the attacker might be able to derive real sources from real message time intervals. We propose the Risk Region (RR) algorithm to secure original message time intervals from UDR attacks. With this algorithm, RR based scheme can effectively increase the attacker's recovery errors, which can be seen from his increased mean square errors.

As future work, we will elaborate on more implementation details, such as the issues of sensor load balance and energy consumption. We will also investigate how to prolong the lifetime of individual sensors and the whole network.

References

[1] Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications. 2003.

[2] Free Haven project, http://freehaven.net/anonbib/topic.html.

[3] Eschenauer L, Gligor VD. A key-management scheme for distributed sensor networks. CCS. 2002.

[4] Chan H, Perrig A, Song D.Random key predistribution schemes for sensor networks. Proc. of IEEE Security and Privacy Symposium. 2003.

[5] Du W, Deng J, Han Y, Varshney P. A pairwise key pre-distribution scheme for wireless sensor networks. CCS. 2003.

[6] Liu D, Ning P. Establishing pairwise keys in distributed sensor networks. CCS. 2003.

[7] Du W, Deng J, Han YS, Chen S, Varshney P. A key management scheme for wireless sensor networks using deployment knowledge. IEEE Infocom. 2004.

[8] Zhu S, Setia S, Jajodia S. Leap: Efficient security mechanisms for large-scale distributed sensor networks. CCS. 2003.

[9] Perrig A, Szewczyk R, Wen V, Culler D, Tygar JD. Spins: Security protocols for sensor networks. Mobicom. 2001.

[10] Li N, Zhang N, Das SK, Thuraisingham B. Privacy preservation in wireless sensor networks: a state-of-the-art survey. Ad Hoc Networks, 2009.

[11] Carbunar B, Yu Y, Shi L, Pearce M, Vasudevan V. Query privacy in wireless sensor networks. SECON. 2007.

[12] Kamat P, Xu W, Trappe W, Zhang Y. Temporal privacy in wireless sensor networks. ICDCS. 2007.

[13] Kamat P, Zhang Y, Trappe W, Ozturk C. Enhancing source-location privacy in sensor network routing. ICDCS. 2005.

[14] Yang Y, Zhu S, Cao G, LaPorta T. An active global attack model for sensor source location privacy: Analysis and countermeasures. SecureComm. 2009.

[15] Yang Y, Shao M, Zhu S, Urgaonkar B, Cao G. Towards event source unobservability with minimum network traffic in sensor networks. ACM WiSec. 2008.

[16] Shao M, Yang Y, Zhu S, Cao G. Towards statistically strong source anonymity for sensor networks. IEEE Infocom. 2008.

[17] Jian Y, Chen S, Zhang Z, Zhang L. Protecting receiver-location privacy in wireless sensor networks. IEEE Infocom, 2007.

[18] Pongaliur K, Xiao L. Maintaining source privacy under eavesdropping and node compromise attacks. IEEE Infocom, 2011.

[19] He W, Liu X, Nguyen H, Nahrstedt K, Abdelzaher T. Pda: Privacy-perserving data aggregation in wireless sensor networks. IEEE Infocom, 2007.

[20] Yang Y, Shao M, Zhu S, Cao G. Towards statistically strong source anonymity for sensor networks. ACM Trans. on Sensor Networks, May 2013.

[21] Axelsson S. The base-rate fallacy and its implications for the difficulty of intrusion detection. CCS. 1999.

[22] Deng J, Han R, Mishra S. Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. DSN. 2004.

[23] Deng J, Han R, Mishra S. Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks. Elsevier Pervasive and Mobile Computing Journal, Special Issue on Security in Wireless Mobile Computing Systems, 2006.

[24] Deng J, Han R, Mishra S. Countermeasures against traffic analysis attacks in wireless sensor networks. Securecomm. 2005.

[25] Ouyang Y, Le Z, Liu D, Makedon F. Source location privacy against laptop-class attacks in sensor networks, SecureComm. 2008.

[26] Mehta K, Liu D, Wright M. Location privacy in sensor networks against a global eavesdropper. ICNP. 2007.

[27] Shao M, Hu W, Zhu S, Cao G, Krishnamurthy S, Porta TL. Cross-layer enhanced source location privacy in sensor networks. SECON. 2009.

[28] Li Y, Ren J. Source-location privacy through dynamic routing in wireless sensor networks. IEEE Infocom. 2010.

[29] Huang Z, Du W, Chen B. Deriving Private Information from Randomized Data. ACM Special Interest Group on Management Of Data (SIGMOD). 2005.

[30] Agrawal R, Srikant R. Privacy-Preserving Data Mining. Special Interest Group on Management Of Data (SIGMOD). 2000.

[31] Wikipedia. Mean Squared Error. http://en.wikipedia.org/wiki/Mean_square_error, 2013.

[32] Zhu Y, Gortler SJ, Thurston D. Sensor Network Localization Using Sensor Perturbation. ACM Trans. on Sensor Networks (TOSN), New York, NY, 4 February 2011.