Data Randomization for Lightweight Secure Data Aggregation in Sensor Network*

Abedelaziz Mohaisen[1], Ik Rae Jeong[2], Dowon Hong[1],
Nam-Su Jho[1], and DaeHun Nyang[3]

[1] Electronics and Telecommunication Research Institute, Daejeon 305-700, Korea
{a.mohaisen,dwhong,nsjho}@etri.re.kr
[2] The Graduate School of Information Security, Korea University, Seoul, Korea
irjeong@korea.ac.kr
[3] Information Security Research Laboratory, Inha University, Incheon, Korea
nyang@inha.ac.kr

Abstract. Data aggregation is one of the main purposes for which sensor networks are developed. However, to secure the data aggregation schemes, several security-related issues have raised including the need for efficient implementations of cryptographic algorithms, secure key management schemes' design and many others. Several works has been introduced in this direction and succeeded to some extent in providing relatively efficient solutions. Yet, one of the questions to be answered is that, can we still aggregate the sensed data with less security-related computation while maintaining a marginal level of security and accuracy? In this paper, we consider data randomization as a possible approach for data aggregation. Since the individual single sensed record is not of a big concern when using data for aggregation, we show how data randomization can explicitly hide the exact single data records to securely exchange them between nodes. To improve the security and accuracy of this approach, we introduce a hybrid scheme that uses the cryptographic approach for a fraction of nodes. We study the efficiency of our schemes in terms of the estimate accuracy and the overhead.

Keywords: security, sensor network, data aggregation, computation efficiency, data randomization, experimental justification.

1 Introduction

Data aggregation is one of the main functions for which the wireless sensor networks (WSN) are developed. In data aggregation networks, the different sensor are scattered in a field for sensing some physical phenomena (e.g., temperature, light, humidity, etc). The avalanched aggregated value of several readings over the time is of more interest rather than the single reading. However, to enable nodes to perform the in-network processing, the concept of secure data aggregation (SDA) is introduced. The SDA has been studied intensively in the context of

* This work was supported by the IT R&D program of MIC/IITA. [2005-Y-001-04, Development of next generation security technology].

F.E. Sandnes et al. (Eds.): UIC 2008, LNCS 5061, pp. 338–351, 2008.

the security study in WSN where several cryptographic-based schemes have been introduced. A nice survey of these works is in [1]. In WSN, the cryptographic-based aggregation schemes are used so far [2,3,4,5]. In these schemes, the sensing node encrypts the raw sensed data using a previously shared key (in the symmetric key model) or public key of the other node (in the public key model) and forwards the encrypted data to the destination which has the corresponding key. The destination in this case is the aggregator. Upon receiving the forwarded encrypted records, the aggregator decrypts them, obtain the raw data, and perform the aggregation function on the aggregated data.

For the above model, both public and symmetric key techniques have been investigated. The public key algorithms have been shown to be computationally feasible to some extent on the typical sensor nodes [6,7,8]. As the public key authentication is crucial requirement for the deployment of public key, authentication services have been introduced in [9,10]. Also, secret key pre-distribution services have been introduced in [11,12,13,14,15] and key revocation techniques have been introduced in [16] to make the applicability of these algorithms and techniques more feasible on the typical sensor nodes. However, to deploy the public key algorithms widely in WSN, several algorithms and designs need to be considered as the aforementioned existing algorithm do not solve the aforementioned problems perfectly. On the other hand, the applicability of symmetric key algorithms in WSN, though computationally feasible, is subject to the *resiliency* and *connectivity* tradeoff [9].

As another direction of performing aggregation, we investigate the applicability of the data randomization for efficient data aggregation. The work is motivated by the question of that: can we still reduce the overhead while performing the same task of *marginally* securing the data aggregation?

To answer the above question, we try the *data randomization* as a solution. The data randomization has been intensively studied in the context of privacy preserving data mining (PPDM) [17]. In the PPDM, the data owner needs to publish an image of his private data to be used by third party for applying data mining algorithms without revealing this data's privacy [18]. That is, the data itself is modified using some mechanisms so that modified data is statistically similar to the original data leading to that some *aggregate functions* can be still applied on the modified data with an acceptable accuracy. An example of the modification techniques is the data perturbation. A promising feature for making the applicability of the randomization more feasible in sensor network is that many randomization components are used already as part of the sensor node design like TinyRNG [19] and RandomLFSR [20]. Though, the data perturbation algorithms face an accuracy/privacy trade-off which is related to that increasing the privacy of the data by increasing the deviation of the added noise (in case of normally distributed noise) results in a high loss in the aggregate accuracy [18,21]. However, one of the facts that may help in reducing the impact of that problem is the huge amount of data delivered by the different sensor nodes over the time making it minimizing the impact of the accuracy loss.

In this paper, we consider the data randomization as a possible technique for secure data aggregation in sensor network. We begin with the randomization-only scenario instead of the existing cryptographic-based aggregation. Facing the accuracy and security problems arising from that, we extend this scheme to more secure/accurate hybrid scheme in which both randomization and cryptographic approaches are utilized. To evaluate our scheme and demonstrate the goal beyond its design, we study the overhead analysis in terms of computation. We also study the accuracy of aggregation estimate.

The rest of this paper is organized as follows: section 2 introduces the assumptions and network models which are used through the paper, section 3 introduces the details of our scheme, section 4 introduces the analysis of our scheme, and finally, section 6 draws concluding remarks for future works.

2 Definitions and Network Model

The nodes in the network are represented as $s_1, s_2, \ldots, s_n$ where the group itself is represented as S. The sensed data by the nodes respectively is denoted with the random variable D where $d_1, d_2, \ldots, d_n \in D$. Also, we define the random variable X which is used to generate noise such that $x_1, x_2, \ldots, x_n \in X$. The above random variable statistical characteristics like the mean and deviation. The mean is $\bar{d}, \bar{x}$ for D and X respectively. Also, we define the noise addition operation $\odot$ which is invertible by $\bar{\odot}$. The following operations and their inference are applied on the random variable realizations:

- $d_{i_{\{\in D\}}} \odot x_{i_{\{\in X\}}} \rightarrow y_{i_{\{\in Y\}}}$. That is, $D \odot X \rightarrow Y$.
- $y_{i_{\{\in Y\}}} \bar{\odot} x_{i_{\{\in X\}}} \rightarrow d_{i_{\{\in D\}}}$. That is, $Y \bar{\odot} X \rightarrow D$.

Through this paper consider the following: D represents the the sensed data, X represents the noise and Y represents the randomized data. In the following, we define the set of definition used through the rest of the paper.

2.1 Definitions

Definition 1 (aggregation function). *For a set of sensed data* $(d_1, d_2, \ldots, d_n) \in D$ *that is sensed by the set of sensor nodes* $s_1, s_2, \ldots, s_n$*, in the context of this paper, the aggregation function* $f(d_1, d_2, \ldots, d_n)$ *is a function that computes a single value result from a collection of inputs. Here, we mainly define the following aggregate function instances:*

- summation*:* $f(d_1, d_2, \ldots, d_n) = \sum_{i=1}^{n} d_i$.
- average*:* $f(d_1, d_2, \ldots, d_n) = \frac{1}{n} \sum_{i=1}^{n} s_i$.
- maximum*:* $f(d_1, d_2, \ldots, d_n) = \max\{d_i | i = 1, 2, \ldots, n\}$
- minimum*:* $f(d_1, d_2, \ldots, d_n) = \min\{d_i | i = 1, 2, \ldots, n\}$
- median*:* $f(d_1, d_2, \ldots, d_n) = d_r : r = \frac{n+1}{2}$ *where* $\{d_1, d_2, \ldots, d_n\}$ *are sorted.*
- count*:* $f(d_1, d_2, \ldots, d_n) = |\{d_i | i = 1, 2, \ldots, n\}|$.

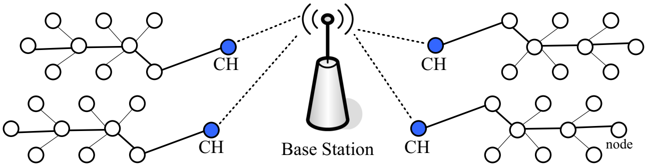


Fig. 1. Illustration: network model

More precisely, in the context of sensor network, the average function is the mostly used.

Definition 2 (distribution function). *Let X be a discrete random variable where $x_1, x_2, \ldots, x_n \in X$. The random variable has a distribution D with a mean value μ and a standard deviation σ. The commutative distribution function (CDF) of X is defined as $F_X(x) = P(X \leq x) = \sum_{x_i \leq x} P(X = x_i) = \sum_{x_i \leq x} p(x_i)$. This function is used to generate all the instances (a.k.a., nonces) of the random variable X.*

Definition 3 (Derandomization). *In the context of this paper and unlike the common use of the word, the derandomization process is defined as the operation of generating a random sequence of random integers that belong to a random variable with specific statistical parameters which are equivalent to the added noise in the randomization phase. Two random variables X and Y are equivalent if they are equal in distribution. That is, $P(X \leq x) = P(Y \leq x) \forall x$.*

2.2 Network Model

In this paper, we use the well known 3-tiers model [22,23]. The network of 3-tiers is a common in the large networks and consists of the huge number of sensor nodes which are classified into clusters [24,25]. The cluster is a functional grouping of the nodes where each cluster has a cluster head (CH). The communication pattern follows a forwarding mechanism where each cluster has backbone nodes which are updated frequently to keep power consumption fair. The cluster head communicates immediately with the base station (BS) or through the sink. The base station can be typically a special kind of sensor node connected to a computer machine. An illustration of the network model is shown in Fig. 1.

3 Randomization for Lightweight Aggregation

In this section, we introduce the details of our scheme including the basic randomization-only scheme, its shortage, and hybrid scheme. Our schemes both consider the above notation and definitions as an underlying intuition.

3.1 Randomization for Secure Data Aggregation

The randomization-only scheme consists of two stages which are namely the offline and online phases. The two phases are performed as follows:

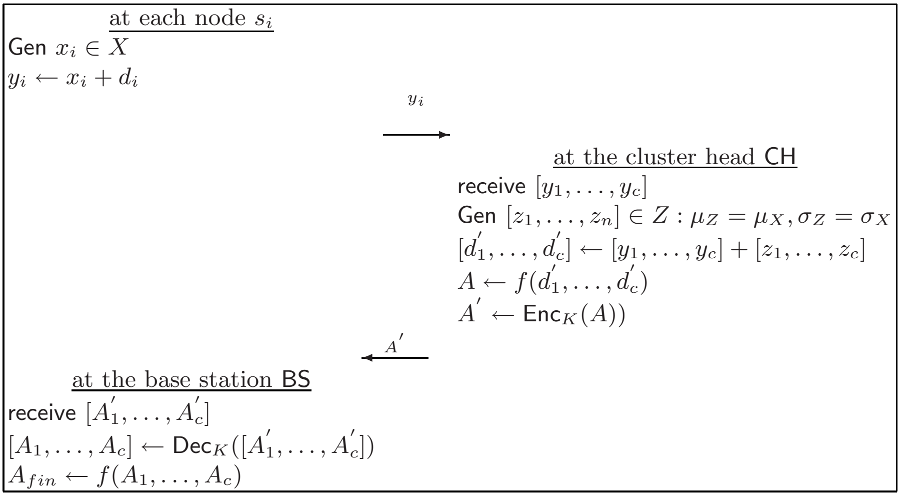
Offline Phase: In the offline phase, the set of node within the cluster agree on the distribution parameters required as entries for the randomization function. That is equivalent to statically assigning the different parameters (e.g., μ, σ, etc) to the different nodes.

Online Phase: In the online phase, the data randomization is performed upon the need of forwarding the data to the cluster head. That is, the procedure interaction in Fig 2 is performed (for cluster with size c). The protocol in Fig. 2 can be summarized as follows:

1. **At Each Node:** each node s_i generates a random nonce $x_i \in X$ using its own parameters, add the generated x_i as a noise to the sensed d_i resulting y_i as $y_i = x_i \odot d_i$, and forwards y_i to the cluster head.
2. **At the Cluster Head:** The cluster head (CH) receives the forwarded randomized data $[y_1, y_2, \ldots, y_c]$ from the different nodes $[s_1, s_2, \ldots, s_c]$ in the cluster, generates a vector of random nonces $[z_1, z_2, \ldots, z_c] \in Z$, remove the equivalent (in distribution) noise to the added one resulting $\widehat{D} = Y \bar{\odot} Z = [y_1, y_2, \ldots, y_c] \bar{\odot} [z_1, z_2, \ldots, z_c]$ where $\bar{\odot}$ is the corresponding items with the corresponding indexes resulting $[d_1^{'}, d_2^{'}, \ldots, d_c^{'}]$. Then, on the resulting $[d_1^{'}, d_2^{'}, \ldots, d_c^{'}]$, the CH performs the aggregation function resulting $A = f(d_1^{'}, d_2^{'}, \ldots, d_c^{'})$ and finally, using some previously shared key K and encryption algorithm Enc, the CH encrypt the resulting A to $A^{'} = \mathsf{Enc}_K(A)$.
3. **At the Base Station (BS):** using some previously agreed-on key K and decryption algorithm Dec, the BS retrieve $A = \mathsf{Dec}_K(A^{'})$ for each received $A^{'}$ value from each CH in the network and then the BS performs its own aggregation function $f(A_1, A_2, \ldots)$ and estimate the final aggregated value.

Note that the aggregation works well because the modification of the data will maintain the same mean of the modified data. That is, the statistical properties of X and Z are same resulting that $E[X + D] = E[Z + D] = E[Y]$ when using the simple addition instead of $\odot$. When using the simple subtraction instead of $\bar{\odot}$ we get that $E[X] = E[Y] - E[Z] = E[Y] - E[X]$.

Limitations: the limitations of the above scheme are two. First, the accuracy due to the randomization grows highly when we set σ to a large enough value that guarantees good standards of security (as shown in Fig. 4(b)). Even though a small deviation could be sufficient for randomization if we set the mean of the noise to some non-zero value, possible data filtering attack can be applied to that once the σ is small. The second shortage is that not all of the aggregation functions shown in Definition 1 can be applied on the perturbed data accurately due to the high variance. For example, the min, max and median functions cannot be applied with the required precision. To overcome this shortage, we introduce the hybrid scheme which introduces reasonable solutions for the two problems.



```
at each node $s_i$
Gen $x_i \in X$
$y_i \leftarrow x_i + d_i$
        $\xrightarrow{y_i}$
                at the cluster head CH
                receive $[y_1, \dots, y_c]$
                Gen $[z_1, \dots, z_n] \in Z : \mu_Z = \mu_X, \sigma_Z = \sigma_X$
                $[d_1^{'}, \dots, d_c^{'}] \leftarrow [y_1, \dots, y_c] + [z_1, \dots, z_c]$
                $A \leftarrow f(d_1^{'}, \dots, d_c^{'})$
                $A^{'} \leftarrow \mathsf{Enc}_K(A))$
        $\xleftarrow{A^{'}}$
at the base station BS
receive $[A_1^{'}, \dots, A_c^{'}]$
$[A_1, \dots, A_c] \leftarrow \mathsf{Dec}_K([A_1^{'}, \dots, A_c^{'}])$
$A_{fin} \leftarrow f(A_1, \dots, A_c)$
```

Fig. 2. The online phase of the randomization-only algorithm

3.2 Hybrid Scheme: Randomization with Encryption

In the above scheme, once the attacker had access to enough number of points in the randomized data, he can study its distribution (when some hint is given on the type of the distribution). Therefore, it would be good choice to harden this kind of natural attack. To do so, the number or randomized data records need to be carefully assigned in that they do not reveal further information. In that case, the attacker will still have some ability to study the distribution but with very high percentage of estimate error.

Our solution for solving this is the hybrid scheme. In this scheme, not only randomization but also encryption is performed. The hybrid scheme also consists of two phases; namely, offline and online phases which are detailed in the following subsections.

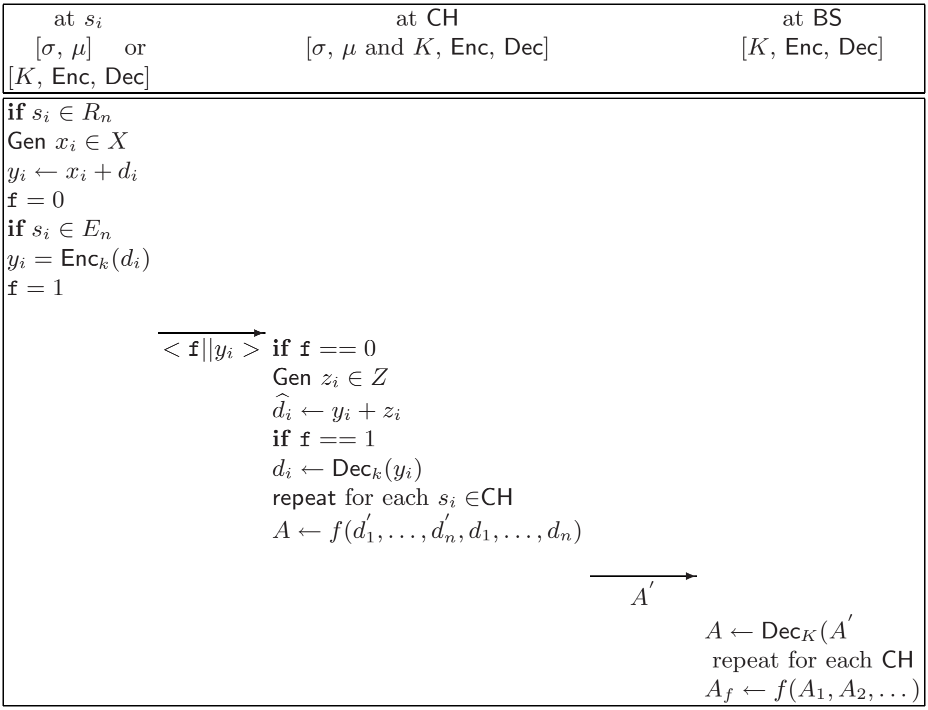
Offline Phase: in this phase, the different nodes in the network are predetermined to whether to use the encryption scheme or the randomization scheme for data delivery. That is, the following is performed:

- The operator divide the sensor nodes to be used within each cluster into two parts representing the number of nodes that will use the encryption scheme and the nodes that will use data randomization. The number of nodes is $n_r = n - n_e$ and n_e for randomization and encryption respectively.
- In the set of nodes to use the randomization scheme, the operator assign the randomization parameters.
- In the set of nodes to use the encryption scheme:
  - The operator assign the encryption scheme to each of the different nodes.
  - Based on the encryption scheme (say, symmetric key model), the operator pre-assign the keys (as in [13]) or keying material (as in [11,12]).

Data Separation: For the packets which include data that has been treated by the randomization method, a '0' flag is attached and for the data which has been encrypted, a '1' flag is added. That is, if $< 0||d_i >$ is received, the cluster head will perform *derandomization* for d_i and if $< 1||d_i >$ is received the CH will use the *decryption*.

Online Phase: In the online phase, the raw data is encrypted or randomized based on the the class of the nodes. That is performed in the following steps.

1. At the node side: For each node s_i in the cluster c, the following is performed on the sensed data item d_i:
   (a) If s_i is in the randomization group, s_i generates a random nonce $x_i \in X$, performs the noise addition to generate y_i as $y_i = x_i \odot d_i$ and forwards $< 0||y_i >$ to the cluster head.
   (b) If s_i is in the encryption group, s_i encrypts d_i using the pre-assigned key and the pre-loaded encryption scheme resulting $y_i = \mathsf{Enc}_K(d_i)$ and forwards $< 1||y_i >$ to the cluster head.
2. At the cluster head: for each received data from the different nodes, the following is performed:



at s_i	at CH	at BS
$[\sigma, \mu]$ or $[K, \mathsf{Enc}, \mathsf{Dec}]$	$[\sigma, \mu$ and $K, \mathsf{Enc}, \mathsf{Dec}]$	$[K, \mathsf{Enc}, \mathsf{Dec}]$

```
at s_i:
    if $s_i \in R_n$
    Gen $x_i \in X$
    $y_i \leftarrow x_i + d_i$
    $\mathtt{f} = 0$
    if $s_i \in E_n$
    $y_i = \mathsf{Enc}_k(d_i)$
    $\mathtt{f} = 1$
                $\xrightarrow{< \mathtt{f}||y_i >}$
at CH:
    if $\mathtt{f} == 0$
    Gen $z_i \in Z$
    $\widehat{d_i} \leftarrow y_i + z_i$
    if $\mathtt{f} == 1$
    $d_i \leftarrow \mathsf{Dec}_k(y_i)$
    repeat for each $s_i \in$CH
    $A \leftarrow f(d_1^{'}, \dots, d_n^{'}, d_1, \dots, d_n)$
                $\xrightarrow{A^{'}}$
at BS:
    $A \leftarrow \mathsf{Dec}_K(A^{'}$
    repeat for each CH
    $A_f \leftarrow f(A_1, A_2, \dots)$
```

Fig. 3. The online phase of the hybrid scheme

(a) If $< 0||y_i >$ is received, using the same randomization parameters at the node side the cluster head CH generates a vector of random nonces $[z_1, z_2, \dots, z_{c-n_r}] \in Z$. Then, the CH performs the addition inverse operation $\bar{\odot}$ for the resulting Z and the received Y resulting $\widehat{D} = Y \bar{\odot} Z \; = [y_1, y_2, \dots, y_{c-n_r}] \; \bar{\odot} [z_1, z_2, \dots, z_{c-n_r}]$ where $\bar{\odot}$ resulting $[d_1^{'}, d_2^{'}, \dots, d_{c-n_r}^{'}]$. Finally, the CH performs the aggregation function on $[d_1^{'}, d_2^{'}, \dots, d_{c-n_r}^{'}]$ that leads to $A_r = f(d_1^{'}, d_2^{'}, \dots, d_{c-n_r}^{'})$.

(b) If $< 1||y_i >$ is received, using some previously agreed-on key K and decryption algorithm Dec, the CH retrieves $d_i = \mathsf{Dec}_K(y_i)$ and performs the aggregate function on the resulting set $d_1, d_2, \dots, d_{n_e}$ resulting $A_e = f(d_1, d_2, \dots, d_{n_e})$.

(c) The CH performs the aggregation function on the results A_e, A_r in the previous two steps resulting $A = f(A_r, A_e)$ and using some previously shared key with the BS (K) and encryption algorithm Enc, the CH encrypt the resulting A resulting $A^{'} = \mathsf{Enc}_K(A)$.

3. At the base station (BS): Using some previously agreed-on key K and decryption algorithm Dec, the BS retrieves $A = \mathsf{Dec}_K(A^{'})$. This is performed for the different cluster heads in the network. Finally, the BS performs its own aggregation function $f(A_1, A_2, \dots)$ and estimate the final aggregated value.

A brief description of this protocol is shown in Fig. 3.

4 Analysis and Evaluation

In this section, we introduce the analysis of our scheme. This typically includes the evaluation of overhead in terms of the required computational power, possible attack scenarios and their countermeasures, and finally the accuracy of aggregation estimate for some commonly used aggregation functions.

4.1 Overhead Evaluation

for the overhead evaluation, there are three scenarios: the randomized only, the encrypted only, and the hybrid scheme. The memory and communication requirements for each scheme is the same however the computation overhead differs.

Scenario 1 (fully randomized): . In the fully randomized scenario, the overall computation overhead results in the computation required the randomization and de-randomization operations at the node and aggregator respectively. That is, $CO = \sum_{i=1}^{n} P_{rand} + \sum_{i=1}^{n} P_{derand} = n(P_{rand} + P_{derand})$. However, P_{rand} is equivalent to P_{derand} (based on definition 3). Assuming that the required computation power for calculating $\odot$ is equal to that for calculating $\bar{\odot}$, the final required computation overhead can be concluded as: $CO = 2\sum_{i=1}^{n} P_{rand} = 2nP_{rand}$. From that, we define the average overhead per node as

$$\overline{CO} = 2\frac{1}{n}\sum_{i=1}^{n} P_{rand} = 2P_{rand}. \tag{1}$$

Scenario 2 (fully encrypted): In the fully encrypted-data scenario, P_e and P_d defines the required power for encryption and decryption respectively. The overhead required computation power can be defined as: $CO = \sum_{i=1}^{n} P_e + \sum_{i=1}^{n} P_d = n(P_e + P_d)$. Similar to scenario 1, we define the average required power as:

$$\overline{CO} = \frac{1}{n}\left(\sum_{i=1}^{n} P_e + \sum_{i=1}^{n} P_d\right) = P_e + P_d. \tag{2}$$

Scenario 3 (hybrid scheme): In the hybrid scheme, both randomization and encryption are used for portions of the network size. Let n_r be the number of randomized and n_e be the number of encrypted and decrypted data. That is, the overall required overhead can be defined as: $CO = \sum_{i=1}^{n_r} P_{rand} + \sum_{i=1}^{n_r} P_{derand} + \sum_{i=1}^{n_e} P_e + \sum_{i=1}^{n_e} P_d = 2\sum_{i=1}^{n-n_e} P_{rand} + \sum_{i=1}^{n_e}(P_e + P_d) = 2(n - n_e)P_{rand} + n_e(P_e + P_d)$. Similarly, we define the average overhead per node as $\overline{CO} = \frac{CO}{n}$ which results:

$$\overline{CO} = \frac{2(n - n_e)P_{rand} + n_e(P_e + P_d)}{n}. \tag{3}$$

By evaluating the above equation at the experimental values for the parameters P_e, P_d and P_{rand}, we can write the average computation (when using TinyRNG) per node as a function of the network size and number of nodes that use the encryption scheme only as: $\overline{CO} = \frac{45.6\,n - 12.72\,n_e}{n}\mu$J.

4.2 Possible Attacks

Several attacks have been studied in the literature of the data perturbation. Some of these attacks are general and some are scheme-specific. However, in all of the attacks regardless to their type, the adversary tries to derive the perturbed data from the modified data given some apostriori knowledge on some of the original data [21,26]. For example, in [26] the independent component analysis technique (ICA) [27] is used to derive the original data from the perturbed data under some conditions. However, this attack will not work with our scheme for two reasons: (i) the data in our scheme is modified separately. (ii) The Non-Gaussianity condition for the original data cannot be satisfied. For similar shortages, the PCA attack in [21] cannot be directly applied to our scheme. Another more serious attack on the additive noise has been also studied in [28]. However, to accomplish a high precision of estimation for the modified data, the deviation σ need to be as small as possible. In our scheme, however, we can set the deviation dynamically considering the required aggregation accuracy and security level.

4.3 Accuracy of Aggregation Estimate

As we previously assign the statistical parameters for the different random variable from which the noise is generated, the resulting aggregate result after the derandomization process will have a small deviation from those values which are calculated on the raw data prior randomization. For example, the deviation of the mean in $\widehat{D}$ from the mean in D is defined as follows:

$$\overline{\Delta d} = |\overline{\widehat{D}} - \overline{d}| = |\frac{1}{n}(\sum_{i=0}^{n} d_i - \sum_{i=0}^{n} d_i^{'})|. \tag{4}$$

In Table 2, the deviation value is used to express the error of the estimate due to the randomization as a percentile from the original prior randomization estimate. That is, these values are expressed as $\frac{\overline{\Delta d}}{\overline{d}} \times 100\%$ for the above mean deviation $\overline{\Delta d}$.

5 Experimental Results

In section, we detail the evaluation of our proposed scheme in terms of the required average overhead in term of computation, the aggregation estimation over the randomized data, and finally, the accuracy of the resulting results compared to those theoretically performed before randomization.

To experimentally estimate the required overhead, we consider the RandomLFSR [20] and TinyRNG [19] as random number generators. For the corresponding symmetric key algorithm, we consider the AES-128.

For evaluating the impact of randomization on the accuracy, we consider Intel Lab Data [1]. The used data reflects sensing four different phenomena which are the voltage, temperature, humidity and light. The data is collected over 32 days using 54 typical sensor nodes. For our usage, we consider a fraction of 1296 readings per node and perform our simulation on them.

5.1 Numerical Results of Power Consumption

The power consumption on the typical Crossbow's Mica2 [29] to perform a randomization for generating a 64-bit random number using the RandomLFSR algorithm [20] is 0.75 μJ [19]. For the same settings, the consumption is 11.4 μJ using the TinyRNG algorithm [19]. The symmetric key's encryption and decryption operations using the AES-128 are estimated at the level of 12.96 μJ and 19.92 μJ respectively [6]. For $n_e = n_r = 50\%$ of n, the overhead in the hybrid scheme can be rewritten as $\overline{CO} = P_{rand} + 0.5(P_e + P_d)$. Based on that, Table 1 is driven. In Table 1, I denotes the saving in the energy as a percentage from the original used in the encryption scheme. That is, I is defined as follows:

$$I = \frac{\overline{CO}_{\text{encryption only}} - \overline{CO}_{\text{specified scheme}}}{\overline{CO}_{\text{encryption only}}} \times 100\%, \tag{5}$$

where the specified scheme can be the hybrid or randomized using either of the randomization algorithms. Though the TinyRNG is computationally heavier than the RandomLFSR resulting a smaller I, the former algorithm (i.e., TinyRNG) is recommended to be used due to its more accurate results [19].

[1] Available at: `http://db.csail.mit.edu/labdata/labdata.html`

Table 1. Comparison between the three scenarios in terms of computation

Protocol	$\overline{CO}$	$\overline{CO}$/RandomLFSR	$\overline{CO}$/TinyRNG
Encryption Only	32.88 μJ	-	-
Randomization Only	-	1.5 μJ ($I = 95.44\%$)	22.8 μJ ($I = 30.66\%$)
Hybrid ($n_e = 50\%$))	-	17.19 μJ ($I = 47.72\%$)	27.84 μJ ($I = 32.39\%$)

5.2 Data Aggregation and Accuracy: Results

To evaluate the accuracy of data randomization, we perform the experiment on the different sensed data records for the above scenario using the same parameters regardless to their values and the values' corresponding interval. Fig. 6 shows a representative plots for the original sensed raw data and Fig. 5 shows the randomized data records. for estimating the accuracy, Table 2 summarizes the aggregation error estimation for the different values. Note that, when using the same σ for the different sensed data regardless to their domain, data records with small interval will be fully distorted and their aggregation accuracy will be low (see Fig. 4(d) and 5(d)). In addition, when σ is relatively small compared to the original data's interval like the case of light aggregation, the distortion will be limited and the accuracy will be high (see 4(c) and 5(c)). To deal with this limitation, σ need to be considered considering the interval of the original data (e.g., with maximum σ as 200% of the mean value of the original data).

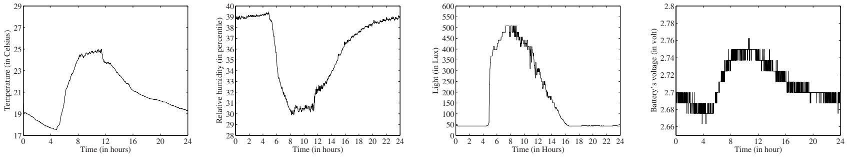
5.3 Impact of Randomization on the Accuracy

To guarantee the minimum standards of security, the deviation σ need to be as high as possible. However, by doing that the accuracy of the aggregated data will be lowered. Fig. 6(a) shows the accuracy of aggregation for the calculated mean over the sensed raw data and Fig 6(b) translates the difference into an accuracy ratio. From the two experiments we figure out that accuracy of the aggregation is proportional with deviation σ. Note that when σ is as big as the original data, the

Table 2. Error estimation in the aggregation results due to data randomization

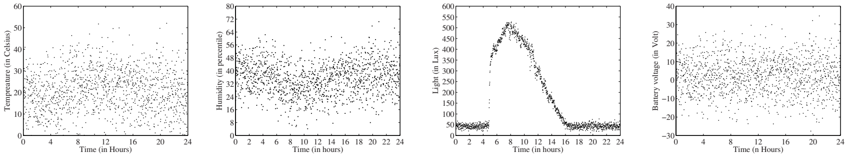
Temperature (noise: $\sigma = 10, \mu = 0$)				Humidity (noise: $\sigma = 10, \mu = 0$)			
Data	Average	Summation	Count	Data	Average	Summation	Count
D	21.0341	27260	1296	D	35.8392	46448	1296
$\widehat{D}$	20.5600	26646	1296	$\widehat{D}$	35.3651	45833	1296
error	2.55%	2.55%	0	error	1.32%	1.32%	0

Light (noise: $\sigma = 10, \mu = 0$)				Voltage (noise: $\sigma = 10, \mu = 0$)			
Data	Average	Summation	Count	Data	Average	Summation	Count
D	177.7460	230360	1296	D	2.7105	3512.8	1296
$\widehat{D}$	177.2719	229740	1296	$\widehat{D}$	2.2364	2898.3	1296
error	0.27%	0.27%	0	error	17.49%	17.49%	0



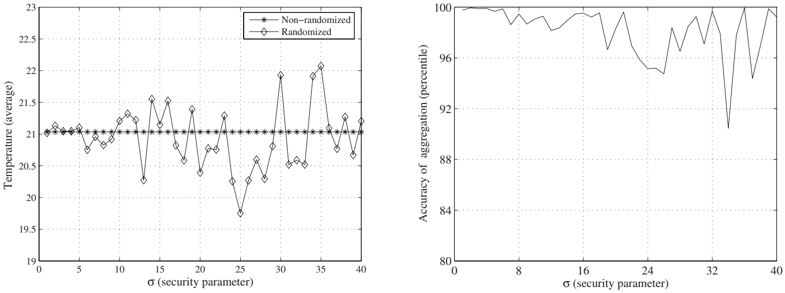
(a) Temperature (b) Humidity (c) Light (d) Voltage

Fig. 4. Raw sensed data over a 24 hours' day from real sensing system representing four different phenomenas from the point of single node



(a) Temperature (b) Humidity (c) Light (d) Voltage

Fig. 5. Randomized versus raw sensed data over a 24 hours day from real sensing system representing four different phenomena from the point of single node



(a) Mean comparison (b) Aggregate Accuracy

Fig. 6. The impact of σ as a security parameter on the accuracy of the aggregation (a) comparison between the non-randomized and randomized aggregation of data for different σ (b) the accuracy of aggregation as a percentile for different σ values

accuracy achieved as higher than 96%. The simulation considers the temperature which can be applied also to the other sensed data with the same consideration.

6 Conclusion and Future Works

The aggregation functions over raw sensed data are meant to perform some statistical functions in which the exact single value is of less importance. In this paper, we utilize this fact and introduce the data randomization as a mean of data hiding. For the attacker to understand the statistical properties of randomized data, he needs to take control over a big fraction of the communication pattern. We showed the efficiency of the randomization in terms of the required computation as the main resources and introduced several perspectives on attacking scenarios including an extension of a hybrid work which generate a trade-off between the resources, accuracy, and security.

In the near future, it will be valuable to study the impact of several statistical distributions on the hardening of the data expectation from the randomized data. Also, we will study the impact of multiple randomizations for the single data records on the accuracy of estimate the security.

Acknowledgment

The authors would like to thank the anonymous reviewers for their comments. Also, they would like to extend their acknowledgment to the hard efforts of Peter Bodik, Wei Hong, Carlos Guestrin, Sam Madden, Mark Paskin, Romain Thibaux, Joe Polastre, and Rob Szewczyk from Intel Lab, Berkeley for making their sensor's data set available online.

References

1. Sang, Y., Shen, H., Inoguchi, Y., Tan, Y., Xiong, N.: Secure data aggregation in wireless sensor networks: A survey. In: PDCAT, pp. 315–320 (2006)
2. Chan, H., Perrig, A., Przydatek, B., Song, D.X.: Sia: Secure information aggregation in sensor networks. Journal of Computer Security 15(1), 69–102 (2007)
3. Chan, H., Perrig, A., Song, D.X.: Secure hierarchical in-network aggregation in sensor networks. In: ACM Conference on Computer and Communications Security, pp. 278–287 (2006)
4. Cam, H., Ozdemir, S., Sanli, H.O., Nair, P.: Secure differential data aggregation for wireless sensor networks
5. Yang, Y., Wang, X., Zhu, S.: Sdap: a secure hop-by-hop data aggregation protocol for sensor networks. In: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing, pp. 356–367 (2006)
6. Wander, A., Gura, N., Eberle, H., Gupta, V., Shantz, S.C.: Energy analysis of public-key cryptography for wireless sensor networks. In: PerCom, pp. 324–328 (2005)
7. Watro, R.J., Kong, D., fen Cuti, S., Gardiner, C., Lynn, C., Kruus, P.: Tinypk: securing sensor networks with public key technology. In: SASN, pp. 59–64 (2004)
8. Malan, D.J., Welsh, M., Smith, M.D.: A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In: First IEEE Int. Conf. on Sensor and Ad Hoc Comm. and Networks, pp. 71–80 (2004)
9. Du, W., Wang, R., Ning, P.: An efficient scheme for authenticating public keys in sensor networks. In: MobiHoc, pp. 58–67 (2005)
10. Nyang, D., Mohaisen, A.: Cooperative public key authentication protocol in wireless sensor network. In: UIC, pp. 864–873 (2006)
11. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: ACM CCS, pp. 52–61 (2003)

12. Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A pairwise key predistribution scheme for wireless sensor networks. ACM Trans. Inf. Syst. Secur. 8(2), 228–258 (2005)
13. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: ACM CCS, pp. 41–47 (2002)
14. Mohaisen, A., Maeng, Y., Nyang, D.: On the grid based key pre-distribution: Toward a better connectivity in wireless sensor networks. In: SSDU, pp. 527–537 (2007)
15. Mohaisen, A., Nyang, D.: Hierarchical grid-based pairwise key pre-distribution scheme for wireless sensor networks. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 83–98. Springer, Heidelberg (2006)
16. Maeng, Y., Mohaisen, A., Nyang, D.: Secret key revocation in sensor networks. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1222–1232. Springer, Heidelberg (2007)
17. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: SIGMOD Conference, pp. 439–450 (2000)
18. Bertino, E., Fovino, I.N., Provenza, L.P.: A framework for evaluating privacy preserving data mining algorithms*. Data Min. Knowl. Discov. 11(2), 121–154 (2005)
19. Francillon, A., Castelluccia, C.: Tinyrng: A cryptographic random number generator for wireless sensors network nodes. In: WiOPT (2007)
20. Lee, N., Philip Levis, J.H.: Mica high speed radio stack (2002)
21. Liu, K., Giannella, C., Kargupta, H.: An attacker's view of distance preserving maps for privacy preserving data mining. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 297–308. Springer, Heidelberg (2006)
22. Arora, A., Ramnath, R., Ertin, E., Sinha, P., Bapat, S., Naik, V., Kulathumani, V., Zhang, H., Cao, H., Sridharan, M., Kumar, S., Seddon, N., Anderson, C., Herman, T., Trivedi, N., Zhang, C., Nesterenko, M., Shah, R., Kulkarni, S.S., Aramugam, M., Wang, L., Gouda, M.G., Choi, Y.-r., Culler, D.E., Dutta, P., Sharp, C., Tolle, G., Grimmer, M., Ferriera, B., Parker, K.: Exscal: Elements of an extreme scale wireless sensor network. In: RTCSA, pp. 102–108 (2005)
23. Dutta, P., Hui, J., Jeong, J., Kim, S., Sharp, C., Taneja, J., Tolle, G., Whitehouse, K., Culler, D.E.: Trio: enabling sustainable and scalable outdoor wireless sensor network deployments. In: IPSN, pp. 407–415 (2006)
24. Bohge, M., Trappe, W.: An authentication framework for hierarchical ad hoc sensor networks. In: WiSe 2003: Proceedings of the 2nd ACM workshop on Wireless security, pp. 79–87. ACM, New York (2003)
25. Shah, R., Roy, S., Jain, S., Brunette, W.: Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. Ad Hoc Networks 1(2-3), 215–233 (2003)
26. Guo, S., Wu, X.: Deriving private information from arbitrarily projected data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 84–95. Springer, Heidelberg (2007)
27. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. In: Proceedings of 15th Conference on Uncertainty in Artificial, vol. 14, pp. 21–30 (2000)
28. Huang, Z., Du, W., Chen, B.: Deriving private information from randomized data. In: SIGMOD Conference, pp. 37–48 (2005)
29. Inc., C.T.: Wireless sensor networks, `http://www.xbow.com/`