A Probabilistic Encryption based MIN/MAX Computation in Wireless Sensor Networks

Bharath K. Samanthula, Wei Jiang and Sanjay Madria
Dept. of Computer Science, Missouri S&T
500 West 15th Street, Rolla, Missouri - 65409
{bspq8, wjiang, madrias}@mst.edu

***Abstract*—Wireless sensor networks (WSNs) have wide range of applications in military, health-monitoring, smart-home applications, and in other commercial environments. The computation of data aggregation functions like MIN/MAX is one of the commonly used tasks in many such WSN applications. However, due to privacy issues in some of these applications, the individual sensor readings should be kept secret from others. That is, the base station should be the only entity who should receive the output of MIN/MAX function and the individual sensor readings should not be revealed either to other sensor nodes or to the root node for confidentiality reasons. Existing Secure Data Aggregation (SDA) techniques for computing MIN/MAX are based on either order preserving or privacy homomorphic encryption schemes which are either inefficient or insecure. Along this direction, this paper proposes two novel solutions for securely computing MIN/MAX functions in WSNs using probabilistic encryption scheme. The first solution works for WSNs with no duplicate sensor readings whereas the second solution acts as a generic method and works even for duplicate readings but is less efficient compared to the first method. However, the second solution is much more secure compared to the existing protocols. The security of the proposed protocols is justified based on the well known quadratic residuosity assumption. We empirically analyze the efficiency of our schemes and demonstrate the advantages of the proposed protocols over existing approaches.**

***Keywords*-Security, Wireless sensor networks, Data aggregation**

I. Introduction

A wireless sensor network (WSN) [1], [2] has promising advantages such as flexibility in deployment, low maintenance and deployment costs [3]. Due to these advantages, WSNs are widely used in many applications such as structural health monitoring [4], military surveillance and reconnaissance [5], [6], wildlife tracking [7], [8], drug administration in hospitals [9], and forest fire detection [10]. Therefore, the computation of data aggregation functions such as COUNT, MIN, MAX, SUM, and AVERAGE in WSN applications are of primary interest. Boundary (minimum and maximum) values are crucial and need to be immediately reported in measurement-critical applications such as fire monitoring systems, radiation level management systems, military reconnaissance and targeting systems. Therefore, in this paper, we restrict our discussion to the secure computation of MIN and MAX functions in a given WSN. However, existing solutions can be utilized to compute other aggregation functions COUNT, SUM, and AVERAGE. More details are given in Section V.

As the sensor nodes are randomly deployed sometimes in inaccessible terrains like in disaster relief operations, the position of a sensor node is not pre-determined. This requires the sensors in WSN to possess self-organizing capabilities. In general, after deployment, the sensor nodes organize themselves into a multi-hop network with the base station as the central point of control. In traditional methods (irrespective of the functionality), the sensed data by each sensor is forwarded, either directly or through intermediate nodes, to the base station through the shortest path. However, as the amount of energy consumption increases with the amount of data transmitted over the network, traditional way of data transmission increases the communication overhead and thereby reduces the overall system lifetime.

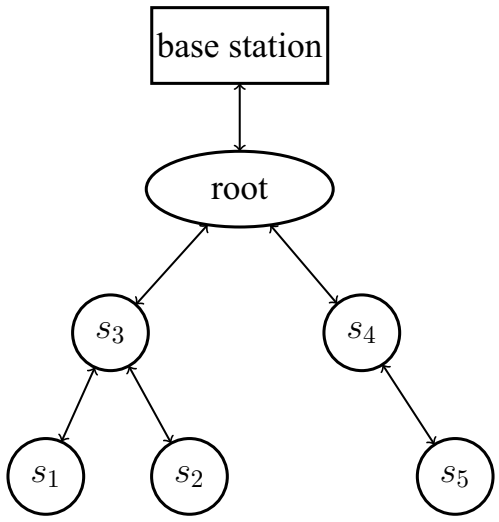

Fig. 1. A multi-hop hierarchical WSN with five sensor nodes

In-network data aggregation combines the data on the fly over the network; therefore, reduces the amount of communication and the energy consumption significantly especially in large sensor networks [11]–[13]. Therefore, in this paper, we assume that WSN is represented as a multi-hop hierarchical[1] sensor network [14]. An example of multi-hop hierarchical

[1]Note that the proposed protocols work for other types of network topologies such as ring topology. That is, the proposed protocols are independent of the network topology.

978-0-7695-4973-6/13 $26.00 © 2013 IEEE
DOI 10.1109/MDM.2013.18

WSN with 5 sensor nodes is shown in Figure 1. A WSN can be logically separated, as shown in Figure 1, into a set of sensor nodes, a root node[2] and the base station. As the sensor nodes have low computation power and low storage capacity, the computation of MIN/MAX functions should be performed in an efficient manner. In addition, when the privacy of sensor nodes is taken into consideration, it is necessary to develop efficient (in terms of computation and communication) Secure Data Aggregation (SDA) methods [15]–[17].

In many privacy preserving applications such as military surveillance, the data sensed by the sensors should be kept private. For example, if the pilot of a warfare helicopter wants to know the minimum or maximum reading of intelligence data about the approximate strength of opposing forces using the WSN, then he/she should only get the corresponding needed information. That is, the sensor node readings other than the minimum or maximum value should not be revealed to the pilot as it may identify an individual/location. In addition, none of the sensor readings should be revealed to the other sensor nodes or to the root node due to the fear of false injection of data or malicious activities. We refer to such a computation as Secure Data Aggregation of minimum and maximum (denoted by SDA_MIN and SDA_MAX respectively).

A naive solution to the SDA_MIN/SDA_MAX problem is for each sensor to directly encrypt his/her reading using a probabilistic and semantically secure encryption scheme and forward the encrypted message to the base station without using in-network aggregation. However, such a simple solution does not achieve the desired goal for two reasons. (i) It reveals all sensor readings to the base station. That is, it leaks more information to the base station than what he/she is authorized to know. (ii) The communication cost between intermediate nodes is proportional to the number of sensor nodes rooted under that sub-tree (i.e., non-constant cost).

This paper proposes two novel SDA_MIN and SDA_MAX protocols based on a probabilistic encryption. Briefly, each of the leaf sensor nodes senses the readings (blood pressure, oxygen level, location data, etc.) and forwards the encrypted data to its parent node who aggregates the encrypted data along with his/her data and forwards the encrypted partially aggregated data to the next level in the tree. This process continues until the root node receives data from all of its children. Then, the root node combines the encrypted data received from its children and forwards the encrypted fully aggregated data to the base station. Finally, the base station decrypts it and identifies MIN/MAX output based on a predetermined condition, more details are given in Section IV.

A. Problem Definition

Let us consider a WSN with n sensor nodes. We assume that the n sensor nodes, the root node, and the base station exist in a tree topology. Without loss of generality, let $s_1, \ldots, s_n$ denote the n sensor readings and let α and β be the minimum and maximum sensor readings respectively. That is,

$$\alpha = \min(s_1, \ldots, s_n) \text{ and } \beta = \max(s_1, \ldots, s_n) \quad (1)$$

The goal of the SDA_MIN (resp., SDA_MAX) protocol is to compute α (resp., β) such that privacy of the sensors is preserved. That is, the individual sensor readings are not revealed to one another. Also, none of the sensor readings is revealed to the root node. In addition, the sensor readings other than α (resp., β) are not revealed to the base station. At the end of the SDA_MIN (resp., SDA_MAX) protocol, the output α (resp., β) is known only to the base station.

B. Our Contributions

Existing SDA_MIN and SDA_MAX algorithms [18], [19] are either inefficient in terms of computation overhead or not secure, can reveal some information. Therefore, in this paper, we propose two novel SDA_MIN and SDA_MAX protocols based on the probabilistic encryption scheme [20]. The contributions of this paper are summarized as follows:

1) **Topology Independent -** As mentioned earlier, we propose two new solutions for solving the SDA_MIN and SDA_MAX problems. The first protocol is suitable for WSNs with no duplicate sensor readings whereas the second protocol works even for duplicate readings. Nevertheless, both protocols are independent of the network topology.
2) **Flexibility -** Though we use a single WSN in this paper, the proposed protocols can be easily extendable for multiple WSNs with the addition of a super root node whose children are root nodes from individual WSNs.
3) **Efficiency -** The proposed protocols are more efficient than the existing work [19]. In addition, the first protocol is more efficient than the second one. In particular, encryption of a bit under the probabilistic encryption scheme used in this paper [20] requires only (at most) three multiplications and one modulo N operation.
4) **Security -** The existing SDA_MIN and SDA_MAX methods do not fully guarantee the privacy of individual sensors. Our first protocol leaks some information to the base station besides the output of the MIN/MAX functionality. However, the second protocol protects the privacy of individual sensors by only revealing the final output to the base station. The security guarantees of the proposed protocols comes from the hardness of quadratic residuosity problem [21], [22].
5) **Generality -** The second protocol proposed in this paper acts as a generic solution and works for WSN with duplicate sensor readings.

The remainder of this paper is organized as follows. Section II summarizes the existing related work. Section III presents some concepts and properties as a background. Section IV discusses the proposed methods in detail. Some directions to securely compute other aggregation functions are discussed in Section V. We present the experimental results in Section VI and conclude the paper with future work in Section VII.

[2]In this paper, we consider the root node as a forwarding node that simply operates on the data received from its children and forwards the new data to the base station. However, the proposed methods can be directly applied to the case where the root node can also sense the data.

II. Related Work

Over the past decade, various in-network data aggregation techniques have been proposed to compute the aggregation functions such as COUNT, SUM, AVERAGE, MIN and MAX in WSNs [23]–[26]. In many sensitive applications such as military surveillance [5], [6], the values of sensor readings should be kept private. Along this direction, much work has been done in combining in-network data aggregation techniques with encryption schemes which led to the evolution of Secure Data Aggregation (SDA) in WSNs [15]–[17].

Assuming a single aggregating node (the root node) and no in-network data aggregation, several SDA algorithms have been proposed to tackle the malicious behavior of nodes [27]–[30] and to compute various aggregation functions [31]. However, sending the individual encrypted sensor readings directly to the root node (i.e., no in-network data aggregation) increases the communication overhead and thereby reduces the overall lifetime of the system. In order to mitigate this effect, researchers have proposed hierarchical SDA algorithms by aggregating the data on the fly over the network using privacy preserving aggregation techniques [17], [32]–[37].

A. Secure Data Aggregation of Minimum and Maximum

In particular to the SDA_MIN and SDA_MAX problems, Acharya et al. [18] proposed a secure comparison of encrypted data at the aggregating nodes in WSNs using order preserving encryption scheme (OPES) [38]. The basic idea is to transform the plain text domain into ciphertext domain by preserving the order. This method leaks much valuable information about individual sensor readings during the comparison of encrypted data at the aggregating nodes. For example, consider a leaf node sending a cipher text c_2 to his/her parent node with cipher text c_1. If $c_1 < c_2$ and the reading of the parent node is 22, then the parent node knows that the actual value sensed by the leaf node is greater than 22. Here, the privacy of leaf node is violated. Hence, their method is not secure.

Ertaul et al. [19] proposed an alternative solution to OPES for calculating the aggregate function MIN/MAX in WSNs using privacy homomorphic encryption schemes [39]–[41]. However, their approach uses a fixed number of pre-defined encrypted values $E(z)$ (where the domain of z depends on the underlying privacy homomorphic scheme used) and $E(0)$ and thus leaks much information such as pattern of sensor readings at the corresponding aggregating nodes. They suggested to fix this problem by re-randomizing the values of $E(z)$ and $E(0)$ by the sensor nodes. However, re-randomization in privacy homomorphic encryption schemes is expensive; therefore, their method has high computational cost at each sensor node.

III. Preliminaries

A. Quadratic Residue

Given a RSA modulus $N = p * q$, where p and q are large primes of similar bit length, the group $\mathbb{Z}_N$ denotes the set of integers $\{0, 1, 2, \ldots, N-1\}$ [42]. The multiplicative group of $\mathbb{Z}_N$ is $\mathbb{Z}_N^* = \{a \in \mathbb{Z}_N \mid gcd(a, N) = 1\}$. In particular, $\mathbb{Z}_N^*$ contains all elements in $\mathbb{Z}_N$ that are co-prime to N. Let $a \in \mathbb{Z}_N^*$. Then, a is said to be a quadratic residue modulo N, or a square modulo N, if there exists an $x \in \mathbb{Z}_N^*$ such that $x^2 \equiv a \mod N$ [21]. If no such x exists, then a is called a quadratic non-residue modulo N. The set of all quadratic residues modulo N is denoted by $\mathbf{Q}_N$ and the set of all quadratic non-residues is denoted by $\overline{\mathbf{Q}}_N$

Example 1. Suppose $N = 21$ and $a = 4$. Here, $a \in \mathbf{Q}_{21}$ since $\exists\ x = 5$ such that $5^2 \equiv 4 \mod 21$. Whereas, $a = 5 \in \overline{\mathbf{Q}}_{21}$ as there exists no $x \in \mathbb{Z}_{21}^*$ such that $x^2 \equiv 5 \mod 21$. □

Definition 1. Given a uniform random element modulo N (with Jacobi Symbol 1), it is believed to be hard to decide whether the element is a square or not unless the factorization of N is known. This is called Quadratic Residuosity Assumption (QRA), a stronger assumption than the factorization [43].

B. Probabilistic Encryption

The probabilistic encryption scheme used in the proposed protocols is introduced by Goldwasser and Micali in [20]. Hereafter, we refer to this encryption scheme as GM. The public key in GM consists of a RSA modulus N and z, where $z \in \mathbb{Z}_N$ is a quadratic non-residue modulo N such that the Jacobi Symbol of z is 1 [42]. To encrypt a single bit b, randomly select $r \in \mathbb{Z}_N^*$, then $E_{GM}(b) = z^b * r^2 \mod N$. The encryption result is a non-quadratic residue if and only if $b = 1$. The security of this encryption scheme is based on the assumption that it is hard to decide quadratic residuosity without knowing the factorization of N. The GM encryption scheme has the following properties [44]:

- XOR-homomorphic property: $E_{GM}(b \oplus b') = E_{GM}(b) * E_{GM}(b') \mod N$, for b, $b' \in \{0, 1\}$
- NOT-property: $E(b \oplus 1) = z * E_{GM}(b) \mod N$
- Re-randomization: Rand($E_{GM}(b)$) = $E_{GM}(b) * E_{GM}(0) \mod N$ is identically distributed to $E_{GM}(b)$

IV. Proposed Solutions

In this section, we propose two new solutions for the SDA_MIN and SDA_MAX problems using the GM encryption scheme under two different assumptions. First, we propose a solution under the assumption that there are no duplicate readings in the WSN. Whereas, the second solution works for all kind of sensor readings[3] and thus can be used as a generic solution. As mentioned earlier, we consider a multi-hop hierarchical WSN with n sensors (whose readings are denoted by $s_1, \ldots, s_n$), the root node and the base station. In addition, for $1 \le i \le n$, we assume that $0 \le s_i \le l$, where l is the domain size for sensor readings. We also assume that the public key (z, N) is known to all nodes in the WSN. However, the private key (p, q) is known only to the base station. For the rest of this paper, α and β represent the minimum and maximum values out of the n sensor readings respectively.

[3]Since encryption supports only integers, fractional values can be easily converted to integers using some scalar factor. For example, a measurement of 4.8 mmol/L blood glucose level by a body implanted sensor can be converted to 48 using a scaling factor of 10.

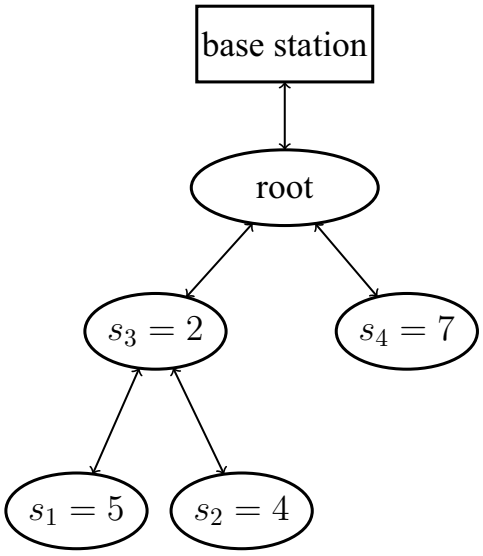

Fig. 2. A sample WSN of four nodes without duplicate sensor readings

A. Approach for WSNs without Duplicate Sensor Readings

Here we present a new approach for solving the SDA_MIN and SDA_MAX problems using the XOR-homomorphic property of GM encryption scheme. We explicitly assume that there exist no duplicate sensor readings in the WSN.

1) The SDA_MIN$_{\text{XOR}}$ Protocol - For $1 \leq i \leq n$, the i^{th} sensor reading s_i is represented by a vector u_i as follows:

$$u_i = \langle \underbrace{0, \ldots, 0}_{s_i}, \underbrace{1, \ldots, 1}_{l-s_i} \rangle \tag{2}$$

Now, we formally define the minimum value of the n sensor readings based on the XOR operation as follows.

Definition 2. Consider n sensor readings $s_1, \ldots, s_n$, and let $u_1, \ldots, u_n$ be their respective vector representations as defined in Equation 2. A new aggregated vector for the minimum, denoted by v, can be computed as $v[j] = u_1[j] \oplus \ldots \oplus u_n[j]$, for $1 \leq j \leq l$. We find the smallest index k such that $v[k] = 1$. Then, the minimum value of the n sensor readings is equivalent to $k - 1$. That is, $\alpha = k - 1$.

Example 2. Consider a WSN with 4 sensor nodes, as shown in Figure 2, whose readings are $s_1 = 5, s_2 = 4, s_3 = 2$, and $s_4 = 7$. Without loss of generality, let $l = 10$. Then, their corresponding vectors, based on Equation 2, are defined as:

$$u_1 = \langle 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 \rangle$$
$$u_2 = \langle 0, 0, 0, 0, 1, 1, 1, 1, 1, 1 \rangle$$
$$u_3 = \langle 0, 0, 1, 1, 1, 1, 1, 1, 1, 1 \rangle$$
$$u_4 = \langle 0, 0, 0, 0, 0, 0, 0, 1, 1, 1 \rangle$$

The new aggregated vector for the minimum is $v = \langle 0, 0, 1, 1, 0, 1, 1, 0, 0, 0 \rangle$. Here, for $k = 3$, we have $v[3] = 1$ and $v[1] = v[2] = 0$. Therefore, the minimum sensor reading $\alpha = k - 1 = 2$ (i.e., reading of node 3 is the minimum). □

Algorithm 1 SDA_MIN$_{\text{XOR}}(s_1, \ldots, s_n) \rightarrow \alpha$

Require: A WSN in tree structure with n sensor nodes - $s_1, \ldots, s_n$ (Note: The private key (p, q) is known only to the base station)

1: Each sensor node i, for $1 \leq i \leq n$:
   (a). Compute vector u_i using s_i based on Equation 2
   (b). Compute $U_i[j] \leftarrow E_{GM}(u_i[j])$, for $1 \leq j \leq l$
2: Each leaf node d, for $1 \leq j \leq l$:
   (a). $M_d[j] \leftarrow U_d[j]$; send $M_d[j]$ to parent node of d
3: Each internal node σ:
   (a). Receive the encrypted partially aggregated vectors M_{w_h} from child node w_h (assuming y child nodes)
   (b). **for** $1 \leq j \leq l$ **do**:
   - **if** σ is not the root **then**
     - $M_\sigma[j] \leftarrow U_\sigma[j] * \Pi_{h=1}^{y} M_{w_h}[j] \bmod N$
     - Send $M_\sigma[j]$ to parent of σ
   - **else**
     - $M_{root}[j] \leftarrow \Pi_{h=1}^{y} M_{w_h}[j] \bmod N$
     - Send $M_{root}[j]$ to the base station

4: base station:
   (a). Receive M_{root}, and decrypt it component-wise to compute k based on Definition 2
   (b). $\alpha \leftarrow k - 1$

Once we know how to compute α using the XOR operations on sensor readings in a WSN, the goal is to achieve this in a secure fashion. Since the basic operation involved is the $\oplus$ (XOR) operation, we utilize the homomorphic property of the GM scheme to design the secure solution. We refer to this solution as SDA_MIN$_{\text{XOR}}$. The overall steps in SDA_MIN$_{\text{XOR}}$ are given in Algorithm 1. Initially, each sensor node compute u_i based on s_i (locally) using Equation 2, encrypts it component-wise i.e., computes $U_i[j] = E_{GM}(u_i[j])$, for $1 \leq j \leq l$. Then, each leaf node d sets $M_d[j]$ to $U_d[j]$ and sends it to his/her parent node (triggered either periodically or upon a query from the base station), for $1 \leq j \leq l$.

Each internal node σ receives the encrypted partially aggregated vectors from the child nodes, aggregates them along with his/her own encrypted vector (if σ is not the root node), and sends the new encrypted aggregated vector M_σ to his/her parent node. Without loss of generality, let us assume that σ has y child nodes denoted by $w_1, \ldots, w_y$ and let $M_{w_1}, \ldots, M_{w_y}$ be their respective encrypted partially aggregated vectors received by σ. Upon receiving, σ computes the new aggregated encrypted vector, for $1 \leq j \leq l$, as follows:

- If σ is not the root node, then compute the j^{th} component of new aggregated encrypted vector as $M_\sigma[j] = U_\sigma[j] * \Pi_{h=1}^{y} M_{w_h}[j] \bmod N$, and send it to his/her parent node.
- Else (i.e., σ is the root node), compute the j^{th} component of the final aggregated encrypted vector M_{root} as $M_{root}[j] = \Pi_{h=1}^{y} M_{w_h}[j] \bmod N$, and send $M_{root}[j]$ to the base station.

TABLE I
DATA TRANSMISSION AT DIFFERENT NODES IN THE SDA_MIN$_{XOR}$ PROTOCOL BASED ON FIGURE 2 FOR $l = 10$

Encrypted Data Transferred by Each Node
node 1 sends $M_1 = \langle E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(1), E_{GM}(1), E_{GM}(1), E_{GM}(1), E_{GM}(1)\rangle$ to node 3 node 2 sends $M_2 = \langle E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(1), E_{GM}(1), E_{GM}(1), E_{GM}(1), E_{GM}(1), E_{GM}(1)\rangle$ to node 3
node 4 sends $M_4 = \langle E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(0), E_{GM}(1), E_{GM}(1), E_{GM}(1)\rangle$ to the root
node 3 sends $M_3 = \langle E_{GM}(0), E_{GM}(0), E_{GM}(1), E_{GM}(1), E_{GM}(0), E_{GM}(1), E_{GM}(1), E_{GM}(1), E_{GM}(1), E_{GM}(1)\rangle$ to the root
root sends $M_{root} = \langle E_{GM}(0), E_{GM}(0), E_{GM}(1), E_{GM}(1), E_{GM}(0), E_{GM}(1), E_{GM}(1), E_{GM}(0), E_{GM}(0), E_{GM}(0)\rangle$ to the base station
base station decrypts M_{root} component-wise, finds $k = 3$, and sets $\alpha = k - 1 = 2$

Finally, the base station decrypts vector M_{root} component-wise and finds the smallest index k, based on Definition 2, to compute α. That is, checks for smallest index k such that $M_{root}[k] \in \overline{\mathbf{Q}}_N$ (denoting a 1). Note that, if a ciphertext belongs to $\mathbf{Q}_N$ (resp., $\overline{\mathbf{Q}}_N$), then it represents an encryption of 0 (resp., 1). The final minimum value α is set to $k-1$ and is known only to the base station.

Example 3. Refer to the sample WSN in Figure 2 and let $l = 10$. Initially, each of the sensor nodes compute their vectors using Equation 2. Then, the leaf nodes 1 and 2 encrypt their vectors u_1 and u_2 component-wise respectively and sends M_1 and M_2 to their parent node 3. The data transmitted by various nodes based on the SDA_MIN$_{XOR}$ protocol are as shown in Table I. Similarly, leaf node 4 sends his component-wise encrypted vector of u_4 (i.e., $U_4 = M_4$) to the root node. Node 3 computes the encrypted partially aggregated vector M_3, for $1 \leq j \leq l$, as follows and sends M_3 to the root node.

$$M_3[j] = U_3[j] * M_1[j] * M_2[j] \bmod N$$

Then, the root computes the final encrypted vector M_{root} as $M_{root}[j] = M_3[j] * M_4[j] \bmod N$, for $1 \leq j \leq l$, and sends it to the base station. Observe that, for $1 \leq j \leq l$:

$$M_{root}[j] = E_{GM}(u_1[j] \oplus u_2[j] \oplus u_3[j] \oplus u_4[4])$$

Finally, the base station decrypts M_{root} component-wise and finds the smallest k based on Definition 2. Here, the decryption of $M_{root}[3]$ gives a value of 1 and also the decryption of $M_{root}[1]$ and $M_{root}[2]$ yields a value of 0. Therefore, the base station finds k as 3 and sets α to $k - 1 = 2$. □

2) The SDA_MAX$_{XOR}$ Protocol - Similar to the SDA_MIN$_{XOR}$ protocol, we propose a new solution to the SDA_MAX problem (denoted by SDA_MAX$_{XOR}$) under the assumption that there exist no duplicate sensor readings in the WSN. Here, we use a different format to generate vector u_i corresponding to the sensor reading s_i which is given as:

$$u_i = \langle \underbrace{1, \ldots, 1}_{s_i}, \underbrace{0, \ldots, 0}_{l-s_i} \rangle \tag{3}$$

Now we formally define the maximum value of the n sensor readings as follows.

Definition 3. For any given n sensor readings $s_1, \ldots, s_n$, let $u_1, \ldots, u_n$ be their respective vector representations as defined in Equation 3. A new aggregated vector for the maximum, denoted by v[4], is computed as $v[j] = u_1[j] \oplus \ldots \oplus u_n[j]$, for $1 \leq j \leq l$. We find the largest index k such that $v[k] = 1$. Then, the maximum value β is equal to k.

Example 4. Consider the WSN with 4 sensors, as shown in Figure 2, whose readings are $s_1 = 5, s_2 = 4, s_3 = 2$, and $s_4 = 7$. Without loss of generality, let $l = 10$. Then, their corresponding vectors, based on Equation 3, are defined as:

$$u_1 = \langle 1,1,1,1,1,0,0,0,0,0\rangle$$
$$u_2 = \langle 1,1,1,1,0,0,0,0,0,0\rangle$$
$$u_3 = \langle 1,1,0,0,0,0,0,0,0,0\rangle$$
$$u_3 = \langle 1,1,1,1,1,1,1,0,0,0\rangle$$

The new aggregated vector for maximum is $v = \langle 0,0,1,1,0,1,1,0,0,0\rangle$. Here, for $k = 7$, we have $v[7] = 1$ and $v[8] = v[9] = v[10] = 0$. Therefore, the maximum sensor reading is $\beta = 7$ (i.e., reading of node 4 is the maximum). □

The main steps involved in SDA_MAX$_{XOR}$ are shown in Algorithm 2. Initially, each of the sensor nodes generate their respective vectors based on Definition 3. Then, each sensor node encrypts his vector u_i component-wise to compute U_i. The process for leaf and internal nodes is the same as in Algorithm 1. Upon receiving the final encrypted vector, the base station decrypts it component-wise and finds the largest index k such that $M_{root}[k] \in \overline{\mathbf{Q}}_N$ (representing a 1).

[4]Note that, depending on the context, we use v to indicate either the aggregated vector for minimum or maximum.

Algorithm 2 SDA_MAX$_{\text{XOR}}(s_1, \ldots, s_n) \rightarrow \beta$

Require: A WSN in tree structure with n sensor nodes - $s_1, \ldots, s_n$ (Note: The private key (p, q) is known only to the base station)

1: Each sensor node i, for $1 \leq i \leq n$:

(a). Compute vector u_i using s_i based on Equation 3

(b). Compute $U_i[j] \leftarrow E_{GM}(u_i[j])$, for $1 \leq j \leq l$

Steps 2 and 3 are same as in Algorithm 1

4. base station:

(a). Receive M_{root}, and decrypt it component-wise to compute k based on Definition 3.

(b). $\beta \leftarrow k$

3) Complexity Analysis - For any given WSN, the computation and communication costs of the SDA_MIN$_{\text{XOR}}$ and SDA_MAX$_{\text{XOR}}$ protocols are the same since the length of the vectors ($= l$), operations at each node, and the data transmitted between the nodes remains the same in both protocols. Therefore, we only analyze the computation and communication costs of the SDA_MIN$_{\text{XOR}}$ protocol. The computation cost is different for a leaf node and an internal node. For a leaf node, the computation cost is determined by step 1(b) of Algorithm 1 where the number of encryptions performed is linearly bounded by the length of the vector. That is, the computation cost of a leaf node is bounded by $O(l)$ number of encryptions.

On the other hand, for each internal node, the computation cost is bounded by $O(l)$ number of encryptions (step 1(b) of Algorithm 1, except for the root node) and $O(y*l)$ number of homomorphic multiplications (step 3(b) of Algorithm 1) where y denotes the number of child nodes to the internal node.

Let K denote the size of the GM encryption key, i.e., length of N in bits (in practice, K should be at least 1024-bit long). As each node transfers a single encrypted vector to his/her parent node, the communication cost between any two nodes is the same and is bounded by $O(K * l)$ in bits (size of each ciphertext is bounded by K bits). Therefore, for a WSN with n sensor nodes, the total communication cost is bounded by $O(n * K * l)$ in bits.

4) Security Analysis - Here, we only analyze the security of the SDA_MIN$_{\text{XOR}}$ protocol. However, a similar security analysis can be deduced for SDA_MAX$_{\text{XOR}}$.

In the SDA_MIN$_{\text{XOR}}$ protocol, due to the probabilistic nature of the GM scheme [20], the messages exchanged between each pair of nodes are random and uniformly distributed in $\mathbb{Z}_N$. Plus, as the private key is known only to the base station, the privacy of individual sensors at each internal node is guaranteed by the quadratic residuosity assumption as explained in Definition 1. This further implies that only the base station can decrypt the final encrypted aggregated vector to compute α. However, in addition to α, the SDA_MIN$_{\text{XOR}}$ protocol leaks the following information to the base station.

- For $k < j \leq l$, if $M_{root}[j] = 1$, then the number of sensors whose reading vectors have '1' at index j are odd. That is, $\sum_{i=1}^{i=n} u_i[j]$ is odd.
- For $k < j \leq l$, if $M_{root}[j] = 0$, then the number of sensors whose reading vectors have '1' at index j are even. That is, $\sum_{i=1}^{i=n} u_i[j]$ is even.

It is our intuition that this information leakage is not useful in deducing any additional information especially in large WSNs.

B. Generic Solution to the Secure Computation of MIN/MAX

Though the above proposed approach is efficient, it works only if the WSN has no duplicate sensor readings, and it also leaks some information. However, in some applications, such as military surveillance [5], duplicate readings exist, and any kind of information leakage is strictly not allowed. Also, sensors with over-lapping reachability areas can sense the same readings. Thus, in order to support duplicate readings and to provide better security, in this sub-section, we propose a new approach for the SDA_MIN and SDA_MAX problems based on the simulation of AND-homomorphic property[5] under GM scheme. This solution acts as a generic method and can be easily extended to other applications upon simple modifications.

As mentioned in [45], the GM encryption scheme can be used to simulate the AND-homomorphic property over $\{0, 1\}$. Let λ be a security parameter and $Z_N^*(+1)$ be the set of elements in Z_N^* with jacobi symbol 1. Bit 1 can be encrypted as a sequence of λ quadratic residues (encryption of λ 0's) and we refer to this as 1-enc. And, bit 0 can be encrypted as a sequence of λ random elements (encryption of λ random bits, with at least one of them being a quadratic non-residue) from $Z_N^*(+1)$, and we refer to this as 0-enc. Let $Encode(b, \lambda, z, N)$ be a function that given a bit b, it produces a corresponding encoding, a vector of length λ. Similarly, let $Decode(x_1, \ldots, x_\lambda, p, q)$ represent the decoding function which takes the encoding (a sequence of λ elements) and private key (p, q) as inputs. The $Decode$ function outputs 1 if all the elements are quadratic residues and 0 if there is at least one quadratic non-residue among the λ elements.

Let $X = \{x_1, \ldots, x_\lambda\}$ and $Y = \{y_1, \ldots, y_\lambda\}$ be the encodings of bits b and b' respectively. Consider the component-wise multiplication of these two encodings $T = \{\tau_1, \ldots, \tau_\lambda\}$, where $\tau_i = x_i * y_i \bmod N$, for $1 \leq i \leq \lambda$. If $b = b' = 1$, then the decoding of T always returns $b \wedge b' = 1$. Also, if exactly one of the bits is 1 (i.e, either $b = 1$ or $b' = 1$), then decoding of T yields $b \wedge b' = 0$. On the other hand, if $b = b' = 0$, then T is an encoding of $b \wedge b' = 0$ with probability $1 - \frac{1}{2^\lambda}$; therefore, decoding of T gives 0 with probability $1 - \frac{1}{2^\lambda}$. Due to the page limitation, we refer the reader to [45] for detailed analysis.

1) The SDA_MIN$_{\text{AND}}$ Protocol - We present a generic approach to the SDA_MIN problem irrespective of whether the WSN has duplicate sensor readings or not.

[5]The simulation of AND gate was proposed in [45], and it was also used in [44] to implement a secure comparison protocol.

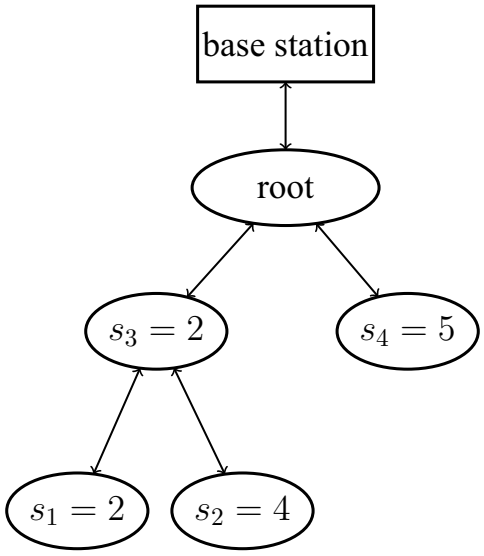

Fig. 3. A sample WSN of four nodes with duplicate sensor readings

Initially, each sensor node i generates the vector u_i, for $1 \leq i \leq n$, based on Equation 3. We formally define the minimum value using the AND operation as follows.

Definition 4. For any given n sensor readings $s_1, \ldots, s_n$, let $u_1, \ldots, u_n$ be their respective vector representations as defined in Equation 3 (observe that, here we use Equations 3 and 2 to compute the vectors for MIN and MAX functionality resp., which is the opposite case in approach 1). A new aggregated vector for the minimum, denoted by v, is computed as $v[j] = u_1[j] * \cdots * u_n[j]$, for $1 \leq j \leq l$. We find the largest index k such that $v[k] = 1$. Then, the minimum value of n sensor readings is defined as $\alpha = k$.

Since the AND operation cannot be computed directly in GM scheme, we will simulate it using the encoding functionality as explained earlier. The overall steps involved in the SDA_MIN$_{\text{AND}}$ protocol are highlighted in Algorithm 3. To start with, each sensor node i using u_i and $Encode()$ function generates the encoding matrix U_i' of size $l \times \lambda$, where each $U_i'[j]$ is the encoding of bit $u_i[j]$ i.e., $U_i'[j] \leftarrow Encode(u_i, \lambda, z, N)$, for $1 \leq j \leq l$. Each leaf node i sets M_i' to his/her encoding matrix and sends it to his/her parent.

As mentioned earlier, let $w_1, \ldots, w_y$ be the child nodes of internal node σ whose encoding matrices are denoted by $M'_{w_1}, \ldots, M'_{w_y}$ respectively. Each of the child node w_h sends his/her encoding matrix M'_{w_h} to the parent node σ. Upon receiving the encoding matrices from all of his/her children, σ performs a component-wise multiplication on all encoding matrices (including his/her own encoding matrix, if σ is not the root node) using XOR-homomorphic property of GM scheme, for $1 \leq j \leq l$ and $1 \leq t \leq \lambda$, as follows:

- If σ is not the root node, then compute the new aggregated encoding matrix as $M'_\sigma[j][t] = U'_\sigma[j][t] * \Pi_{h=1}^{y} M'_{w_h}[j][t] \bmod N$, and send it to his/her parent.
- Otherwise (the case where σ is the root node), σ computes the (j,t) component of the final encoding matrix M'_{root} as $M'_{root}[j][t] = \Pi_{h=1}^{y} M'_{w_h}[j][t] \bmod N$, and sends $M'_{root}[j][t]$ to the base station.

Algorithm 3 SDA_MIN$_{\text{AND}}(s_1, \ldots, s_n) \rightarrow \alpha$

Require: A WSN in tree structure with n sensor nodes - $s_1, \ldots, s_n$ (Note: The private key (p, q) is known only to the base station)

1: Each sensor node i, for $1 \leq i \leq n$:
   (a). Compute vector u_i using s_i based on Equation 3
   (b). $U_i'[j] \leftarrow Encode(u_i, \lambda, z, N)$, for $1 \leq j \leq l$
2: Each leaf node d, for $1 \leq j \leq l$ and $1 \leq t \leq \lambda$:
   (a). $M_d'[j][t] \leftarrow U_d'[j][t]$; send $M_d'[j][t]$ to parent of d
3: Each internal node σ:
   (a). Receives the encoding matrices M'_{w_h} from child node w_h (assuming y child nodes)
   (b). **for** $1 \leq j \leq l$ and $1 \leq t \leq \lambda$ **do**:
   - **if** σ is not the root **then**
     – $M'_\sigma[j][t] \leftarrow U'_\sigma[j][t] * \Pi_{h=1}^{y} M'_{w_h}[j][t] \bmod N$
     – Send $M'_\sigma[j][t]$ to parent of σ
   - **else**
     – $M'_{root}[j][t] \leftarrow \Pi_{h=1}^{y} M'_{w_h}[j][t] \bmod N$
     – Send $M'_{root}[j][t]$ to the base station
4: base station:
   (a). Receive M'_{root}, and decode it row-wise to compute k based on Definition 4
   (b). $\alpha \leftarrow k$

Upon receiving M'_{root} from the root, the base station decodes each row vector (i.e., computes $Decode(M'_{root}[j], p, q)$ for $1 \leq j \leq l$) and finds the maximum row index k such that $Decode(M'_{root}[k], p, q) = 1$. Finally, the base station sets the minimum value as $\alpha = k$. As mentioned earlier, $Decode(M'[k], p, q) = 1$ if and only if $M'[k][t] \in \mathbf{Q}_N$, $\forall\ 1 \leq t \leq \lambda$. Similarly, $Decode(M'[j], p, q) = 0$ if and only if $\exists\ t$ such that $M'[k][t] \in \overline{\mathbf{Q}}_N$, for $1 \leq t \leq \lambda$.

Example 5. Refer to Figure 3 with duplicate sensor readings and let $l = 6$ and $\lambda = 5$. Initially, each of the sensor nodes compute their vectors based on Equation 3. Then, each sensor node generates his/her encoding matrix U_i' using vector u_i. Consider node 1, we have $u_1 = \langle 1, 1, 0, 0, 0, 0 \rangle$ and the corresponding encoding matrix is $U_1' = \langle$ 1-enc, 1-enc, 0-enc, 0-enc, 0-enc, 0-enc $\rangle$. Where 1-enc and 0-enc are the encodings of bits 1 and 0 respectively. The data transmitted by different nodes based on the SDA_MIN$_{\text{AND}}$ protocol are as shown in Table II. Upon receiving the encoding matrix M'_{root} from the root, the base station decodes each row-vector of M'_{root} and finds the largest index k. Following from Table II, the decoding of $M'_{root}[2]$ gives a value of 1 and also the decoding of $M'_{root}[3], M'_{root}[4], M'_{root}[5]$ and $M'_{root}[6]$ yields a value of 0. Therefore, the base station sets $k = \alpha = 2$. □

TABLE II
DATA TRANSMISSION AT DIFFERENT NODES BASED ON THE SDA_MIN$_{AND}$ PROTOCOL

Encrypted Data Transferred by Each Node
node 1 sends $M'_1 = \langle$ 1-enc, 1-enc, 0-enc, 0-enc, 0-enc, 0-enc $\rangle$ to node 3 node 2 sends $M'_2 = \langle$ 1-enc, 1-enc, 1-enc, 1-enc, 0-enc, 0-enc $\rangle$ to node 3
node 4 sends $M'_4 = \langle$ 1-enc, 1-enc, 1-enc, 1-enc, 1-enc, 0-enc $\rangle$ to the root
node 3 sends $M'_3 = \langle$ 1-enc, 1-enc, 0-enc, 0-enc, 0-enc, 0-enc $\rangle$ to the root
root sends $M'_{root} = \langle$ 1-enc, 1-enc, 0-enc, 0-enc, 0-enc, 0-enc $\rangle$ to the base station
base station decrypts each row-vector of M'_{root} and finds k as 2 and sets $\alpha = k = 2$

Algorithm 4 SDA_MAX$_{AND}(s_1, \ldots, s_n) \rightarrow \beta$

Require: A WSN in tree structure with n sensor nodes - $s_1, \ldots, s_n$ (Note: The private key (p, q) is known only to the base station)

1: Each sensor node i, for $1 \leq i \leq n$:
   (a). Computes vector u_i using s_i based on Equation 2
   (b). $U'_i[j] \leftarrow Encode(u_i, \lambda, z, N)$, for $1 \leq j \leq l$

Steps 2 and 3 are same as in Algorithm 3

4. base station:
   (a). Receive M'_{root}, decode it row-wise, and compute the smallest index k such that $Decode(M'[k], p, q) = 1$
   (b). $\beta \leftarrow k - 1$

2) The SDA_MAX$_{AND}$ Protocol - Similar to the SDA_MIN$_{AND}$ protocol, we present a new approach to the SDA_MAX problem (denoted by SDA_MAX$_{AND}$) based on the simulation of AND-homomorphic property of GM scheme. Upon a query from the base station, each of the sensor nodes generate their respective vectors u_i, using s_i, based on Equation 2. Using u_i and the $Encode()$ function, the i^{th} sensor node generates an encoding matrix U_i as explained earlier. The rest of the steps for leaf and internal nodes are same as in Algorithm 3. Once the base station receives the final encoding matrix M'_{root}, it decodes each row-vector of M'_{root} and finds the smallest index k such that $Decode(M'_{root}[k], p, q) = 1$. The overall steps involved in the SDA_MAX$_{AND}$ protocol are given in Algorithm 4. Finally, the base station sets the maximum value as $\beta = k - 1$.

3) Complexity Analysis - For any given WSN, as the computation and communication costs for SDA_MIN$_{AND}$ and SDA_MAX$_{AND}$ are the same, we only analyze the complexity of SDA_MIN$_{AND}$. For a leaf node, the computation cost is determined by step 1(b) of Algorithm 3 where the number of encryptions performed is linearly bounded by the size of the encoding matrix. That is, the computation cost of a leaf node is bounded by $O(l*\lambda)$ number of encryptions. On the other hand, for an internal node, the computation cost is bounded by $O(l*\lambda)$ number of encryptions (step 1(b) of Algorithm 3, except for the root node) and $O(y*l*\lambda)$ number of homomorphic multiplications (step 3(b) of Algorithm 3) where y denotes the number of child nodes to the internal node. The computation cost of base station is bounded by $O(l*\lambda)$ decryptions.

Furthermore, the communication cost between any two nodes is the same and is bounded by $O(K*l*\lambda)$ in bits. Therefore, for a WSN with n sensor nodes, the total communication cost is bounded by $O(n*K*l*\lambda)$ in bits.

4) Security and Correctness Analysis - In the SDA_MIN$_{AND}$ protocol, as the internal nodes do not have the private key, they cannot decode the encoding matrices and hence the privacy of sensors is preserved at the internal nodes. On the other hand, only the base station can decrypt the final encoding matrix to compute α. In the SDA_MIN$_{AND}$ protocol, decoding of the row vectors with index greater than k will always yield a value of 0 with probability $1 - \frac{1}{2^\lambda}$; therefore, the chances of getting a wrong result is negligible. A similar analysis can be deduced for SDA_MAX$_{AND}$. In practice, for $\lambda = 30$, the probability of retrieving the correct and desired output by the base station is $1 - \frac{1}{2^{30}} \approx 1 - 10^{-9} \approx 1$.

V. OTHER AGGREGATION FUNCTIONS

In this paper, two new approaches have been proposed for securely computing the aggregates MIN and MAX in a given WSN. A natural question that arises is whether these approaches support other aggregates such as COUNT, SUM, and AVERAGE. We observe that the proposed protocols cannot be extended to support aggregates other than MIN and MAX. However, we emphasize that additive homomorphic schemes, such as Elliptic Curve Elgamal [46], can be utilized for this purpose. For example, recent work in [17] shows how to securely compute the aggregate function SUM in WSNs. Nevertheless, more research needs to be done to develop a

TABLE III
COMPARISON OF THE PROPOSED PROTOCOLS WITH EXISTING WORK

Method	Encryption Time	Energy Consumption
Acharya et al. [18]	170 ms	0.918 mJ
SDA_MIN$_{XOR}$	1676 ms	9.050 mJ
SDA_MIN$_{AND}$	50300 ms	271.620 mJ

generic secure framework that is energy-efficient (both in terms of computation and communication) and supports all aggregation functions which are very efficient computationally.

VI. EMPIRICAL ANALYSIS

In this section, we empirically compute the efficiency of the proposed protocols and compare it with the existing methods.

All protocols were implemented in nesC [47] and tested on a TelosB sensor mote with 10K byte RAM running TinyOS 1.1.10. We fix the values of l, λ, and the encryption key size (i.e., K) to 20, 30, and 1024 bits respectively for the rest of this section (similar analysis can be deduced for other values). Since the computation and communication costs of SDA_MIN$_{XOR}$ (resp., SDA_MIN$_{AND}$) and SDA_MAX$_{XOR}$ (resp., SDA_MAX$_{AND}$) are the same, we only present the costs of SDA_MIN$_{XOR}$ and SDA_MIN$_{AND}$.

We first compare the encryption costs of a sensor in the proposed protocols with the OPES-based protocol of Acharya et al. [18]. The results are as shown in Table III. It is clear that the proposed protocols are much costlier than their work. More specifically, for a given sensor node, the encryption costs are 1676 and 50300 milliseconds for SDA_MIN$_{XOR}$ and SDA_MIN$_{AND}$ respectively. Also, it took only 170 milliseconds for the method in [18] since it is based on non-cryptographic primitives. However, as explained in Section II, their method is not secure; because, it reveals the relative ordering of the sensor readings to the internal nodes which is treated as a violation of privacy of the corresponding sensor.

In addition, the energy consumption of a sensor (based on the encryption cost) in the proposed protocols is computed. As shown in Table III, the energy consumptions are 9.050 and 271.620 millijoules for SDA_MIN$_{XOR}$ and SDA_MIN$_{AND}$ respectively. On the other hand, the energy consumption for a sensor in the OPES based scheme [18] is 0.918 millijoules. We emphasize that our second protocol, which acts as a generic solution, is more secure than [18], but this advantage comes at the expense of extra energy consumption i.e., 270.702 millijoules more compared to [18]. In general, each TelosB mote is powered by 2 AA batteries, and each battery has 11050 joules. Therefore, based on 271.620 mJ per use, each sensor will survive enough for any given application based on the proposed protocols.

On the other hand, the existing method in [19] is based on the privacy homomorphism and is not secure against plaintext attacks (this issue was also mentioned in [19]). To overcome this issue, the authors in [19] suggested to replace privacy homomorphism with additive homomorphic schemes such as Paillier cryptosystem [48]. Nevertheless, we observe that Paillier cryptosystem is much more costlier than the proposed protocols. Note that encryption in the GM scheme, upon which the proposed protocols are constructed, involves only at most 3 multiplications and one modulo N operation. Where as in Paillier scheme, an encryption requires roughly $2\log_2 N$ multiplications and one modulo N^2 operation. Specifically, the Paillier based version of [19] takes almost 132 and 37 times more computational time compared to SDA_MIN$_{XOR}$ and SDA_MIN$_{AND}$ respectively. A similar trend can be observed for other additive homomorphic schemes.

In general, security and efficiency/energy consumption are two opposite goals and one has to trade-off between the two depending on the application requirements. We emphasize that the main goal of this paper is to develop secure aggregating protocols for MIN and MAX in WSNs which are also efficient over the existing method [19]. Based on the above discussions, it is clear that our second protocol, i.e., SDA_MIN$_{AND}$ provides higher security (based on quadratic residuosity assumption) than [18], and is also more efficient than Paillier-based version of [19]. Furthermore, SDA_MIN$_{XOR}$ is more efficient than SDA_MIN$_{AND}$ (almost by a factor of λ), and is more suitable for WSN applications that guarantees no duplicate sensor readings.

VII. CONCLUSIONS AND FUTURE WORK

WSNs offer various advantages [3] in different areas such as military, environment, health, home, and other commercial applications. In many such applications, however, when the privacy of a sensor's data is taken into consideration, developing efficient secure data aggregation protocols for MIN and MAX (denoted by SDA_MIN & SDA_MAX) becomes more challenging. Along this direction, we proposed two new approaches for SDA_MIN and SDA_MAX problems based on the probabilistic encryption scheme proposed by Goldwasser and Micali (GM) [20]. The first approach is based on the XOR-homomorphic property whereas the second approach utilizes the simulation of AND gate under GM scheme.

The first approach, though more efficient than the Paillier based version of [19], works only if the sensor nodes have no duplicate readings, and also leaks some information. However, the second approach acts as a generic method and is also more secure than the first approach and OPES based scheme [18]. This advantage comes with additional computational cost and thus, the second approach is less efficient compared to the first one. Nevertheless, both of the proposed methods are efficient compared to the Paillier based solution of [19]. Depending on the application requirements, the network administrator can choose between the two approaches thereby balancing the trade-off between efficiency and security.

The efficiency of the second approach depends on the size of the encoding matrix (i.e., $l \times \lambda$) at each node. As a future work, we will improve the efficiency of the generic method by reducing the size of the encoding matrix using heuristics and dimensionality reduction techniques. Also, we will investigate alternative methods for other aggregation functions.

ACKNOWLEDGMENT

This material is based upon work supported by the Office of Naval Research under Award No. N000141110256 and a grant from Army Research Lab.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102 – 114, August 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer and Telecommunications Networking*, vol. 52, pp. 2292–2330, 2008.

[3] C.-Y. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[4] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Wireless sensor networks for structural health monitoring," in *Proceedings of ACM SenSys*, 2006, pp. 427–428.

[5] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications : Remote large-scale environments," in *IEEE Military Communications Conference (MILCOM)*, October 2009, pp. 1–7.

[6] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *Proceedings of ACM SenSys*, 2004, pp. 1–12.

[7] J.-H. Huang, Y.-Y. Chen, Y.-T. Huang, P.-Y. Lin, Y.-C. Chen, Y.-F. Lin, S.-C. Yen, P. Huang, and L.-J. Chen, "Rapid prototyping for wildlife and ecological monitoring," *IEEE Systems Journal*, vol. 4, no. 2, pp. 198–209, 2010.

[8] P. Sikka, P. Corke, and L. Overs, "Wireless sensor devices for animal tracking and control," in *the 29th Annual IEEE International Conference on Local Computer Networks*, november 2004, pp. 446–454.

[9] J. Ko, C. Lu, M. Srivastava, J. Stankovic, A. Terzis, and M. Welsh, "Wireless sensor networks for healthcare," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, November 2010.

[10] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," in *Proc. of IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, 2005, pp. 1214–1217.

[11] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of International Conference on Distributed Computing Systems*, 2002, pp. 575–578.

[12] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70–87, April 2007.

[13] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey," *Comm. Surveys & Tutorials, IEEE*, vol. 8, pp. 48–63, 2006.

[14] C. Buratti and R. Verdone, "A hybrid hierarchical multi-hop wireless network: From wireless sensors to the fixed infrastructure," in *IEEE MASS*, 2007, pp. 1–6.

[15] H. Alzaid, E. Foo, and J. G. Nieto, "Secure data aggregation in wireless sensor network: a survey," in *Proceedings of Australasian conference on Information security*. Australian Computer Society, 2008, pp. 93–105.

[16] V. Kumar and S. Madria, "Secure data aggregation in wireless sensor networks," in *Wireless Sensor Network Technologies for the Information Explosion Era*. Springer-Verlag, 2010, pp. 77–107.

[17] V. Kumar and S. Madria, "Secure hierarchical data aggregation in wireless sensor networks: Performance evaluation and analysis," in *MDM*. IEEE Computer Society, 2012, pp. 196–201.

[18] M. Acharya, J. Girao, and D. Westhoff, "Secure comparison of encrypted data in wireless sensor networks," in *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WIOPT)*, april 2005, pp. 47–53.

[19] L. Ertaul and V. Kedlaya, "Computing aggregation function minimum/maximum using homomorphic encryption schemes in wireless sensor networks," in *ICWN'07*, 2007, pp. 186–192.

[20] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and Systems Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[21] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007, ch. Additional Public-Key Encryption Schemes, pp. 385–416.

[22] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *EUROCRYPT'02*, vol. 2332. Springer-Verlag, May 2002, pp. 45–64.

[23] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," in *Symposium on Operating Systems Design and Implementation*, 2002, pp. 131–146.

[24] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," in *ICDE*, 2004.

[25] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *Proceedings of ACM SenSys*, 2004, pp. 250–262.

[26] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *ACM SIGMOD Record*, vol. 31, pp. 9–18, 2002.

[27] D. Wagner, "Resilient aggregation in sensor networks," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, ser. SASN '04, 2004, pp. 78–87.

[28] L. Buttyán, P. Schaffer, and I. Vajda, "Resilient aggregation with attack detection in sensor networks," in *IEEE International Conference on Pervasive Computing and Communications Workshops*, march 2006.

[29] L. Buttyán, P. Schaffer, and I. Vajda, "Ranbar: Ransac-based resilient aggregation in sensor networks," in *the fourth ACM workshop on Security of ad hoc and sensor networks*, ser. SASN '06, 2006, pp. 83–90.

[30] A. Mahimkar and T. Rappaport, "Securedav: A secure data aggregation and verification protocol for sensor networks," in *Proceedings of IEEE Global Telecommunications Conference*, 2004.

[31] B. Przydatek, D. Song, and A. Perrig, "Sia: secure information aggregation in sensor networks," in *Proceedings of ACM SenSys*, 2003, pp. 255–265.

[32] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks," in *ACM CCS*, 2006, pp. 278–287.

[33] J. Girao, M. Schneider, and D. Westhoff, "CDA: Concealed data aggregation in wireless sensor networks," in *ACM Workshop on Wireless Security*, 2004.

[34] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, july 2005, pp. 109–117.

[35] H. Çam, S. Özdemir, P. Nair, D. Muthuavinashiappan, and H. Ozgur Sanli, "Energy-efficient secure pattern based data aggregation for wireless sensor networks," *Computer Communications*, vol. 29, pp. 446–455, February 2006.

[36] S.-I. Huang, S. Shieh, and J. D. Tygar, "Secure encrypted-data aggregation for wireless sensor networks," *Journal of Wireless Networks*, vol. 16, no. 4, pp. 915–927, May 2010.

[37] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure data aggregation in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1040–1052, 2012.

[38] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *ACM SIGMOD*, 2004, pp. 563–574.

[39] J. D. I. Ferrer, "A new privacy homomorphism and applications," *Information Processing Letters*, vol. 60, pp. 277–282, December 1996.

[40] J. Domingo-ferrer and J. Herrera-Joancomarti, "A privacy homomorphism allowing field operations on encrypted data," in *I Jornades de Matematica Discreta i Algorismica*, 1998.

[41] J. Domingo-Ferrer, "A provably secure additive and multiplicative privacy homomorphism," in *ISC*. Springer-Verlag, 2002, pp. 471–483.

[42] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*. CRC Press, Inc., 1996, ch. Mathematical Background.

[43] D. Hofheinz and E. Kiltz, "The group of signed quadratic residues and applications," in *CRYPTO*. Springer-Verlag, 2009, pp. 637–653.

[44] M. Fischlin, "A cost-effective pay-per-multiplication comparison method for millionaires," in *CT-RSA*. Springer-Verlag, 2001, pp. 457–472.

[45] T. Sander, A. Young, and M. Yung, "Non-interactive cryptocomputing for nc1," in *FOCS*. IEEE Computer Society, 1999, pp. 554–566.

[46] E. Mykletun, J. Girao, and D. Westhoff, "Public key based cryptoschemes for data concealment in wireless sensor networks," in *IEEE ICC*, vol. 5, 2006, pp. 2288 –2295.

[47] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: A holistic approach to networked embedded systems," in *Proceedings of the ACM SIGPLAN*, 2003, pp. 1–11.

[48] P. Paillier, "Public key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*. Springer-Verlag, 1999, pp. 223–238.