Privacy-Preserving MAX/MIN Query Processing for WSN-as-a-Service

Hua Dai*†, Yan Ji*, Fu Xiao*, Geng Yang*†, Xun Yi‡, Lei Chen*†
* Nanjing University of Post and Telecommunication, Nanjing, China, 210023
†Jiangsu Security and Intelligent Processing Lab of Big Data, Nanjing, China, 210023
‡Royal Melbourne Institute of Technology University, Melbourne, Australia, VIC 3000

***Abstract*—WSN-as-a-Service (WaaS) is a novel application model of wireless sensor networks (WSNs). Owners of WSNs provide data queries as services, while users pay for needed services as they use such services. The adoption of WaaS improves the usage of WSNs and reduces the cost of network deployment and maintenance. It is challenging to protect data from curious users while, at the same time, providing MAX/MIN query services. In this paper, we propose a privacy-preserving MAX/MIN query processing method for WaaS. To the best of our knowledge, this work is the first to discuss a privacy-preserving data query method in the WaaS environment. To implement privacy-preserving MAX/MIN queries, we propose a novel query protocol by adopting the idea of secure multi-party computation. The protocol consists of two cooperative query processing algorithms that are deployed in the aggregate sensor and normal sensors. During query processing, multiple rounds of secure interactions between sensors are performed. In each round, one bit of the query result is determined through cooperation of sensors, while the data of sensors participating in query processing remain private. Curious users cannot obtain any private data from the network even if a few compromised sensors collude with them. The analysis and evaluations indicate that the proposed protocol computes query results reliably, aviods the energy hole problem and is efficient in terms of communication cost.**

***Index Terms*—Wireless sensor network, WSN-as-a-Service, Privacy-preserving, MAX/MIN query**

I. Introduction

Wireless sensor networks, as one of the key technologies of the IoT [1], have been applied in various important areas, such as medicine, environmental monitoring, national defense, etc. However, privacy disclosure is becoming increasingly critical in the applications of WSNs. For example, in the healthcare domain, patients' private medical data could be stolen for drug marketing; in smart home applications, the data on the use of water and electricity in daily life could be intercepted and used to commit theft. As a result, privacy protection is becoming a hot issue in the research and applications of wireless sensor networks.

Nowadays, X-as-a-Service is a very popular and practical methodology in IT whereby a resource "X" is provided as an on-demand service, such as Software-as-a-Service (SaaS), etc. Similarly, WSNs can be treated as a service which we refer to as WSN-as-a-Service (WaaS). In WaaS, owners of the deployed WSNs provide data access services, such as data query, status monitoring, etc., for additional profit. Users can pay for needed services as they use such services, which reduces their costs by eliminating the need for WSNs deployment and maintenance. By adopting WaaS, the usage of WSNs is improved, benefiting both owners and users.

ISBN 978-3-903176-16-4 ©2019 IFIP

MAX/MIN query is a useful data query for obtaining the maximum or minimum of data in the area and time period of interest. For example, it can be used to monitor the highest temperature in a warehouse for fire alarms. In WaaS, the result of a MAX/MIN query, contributed by a sensor, is public to users once they have paid for the service. However, the data collected by other sensors are private and should be protected from users. In the curious-but-honest model [2], users are curious about that private data and attempt to obtain them but will strictly follow the established protocols. It is a challenge to protect private data from "curious" users while simultaneously providing MAX/MIN query services.

In this paper, we propose a privacy-preserving MAX/MIN query processing method in a WaaS environment by adopting the idea of secure multi-party computation [6]. Detailed processing procedures are illustrated for the given privacy-preserving MAX/MIN query protocol (PMQ) that deploys two cooperative query processing algorithms in the aggregate sensor and normal sensors. The aggregate sensor is a sensor in a WSN that acts as the bridge between users and the network. Once a query service has been paid for and started, owners of WSNs first generate and configure secure codes for sensors in the network that is utilized to maintain the privacy of collected data involved in query processing. Then, multiple rounds of interactions between normal sensors and the aggregate sensor are performed to process the query. In each round, one bit of the query result is determined by the cooperation of all sensors in the network, following the secure multi-party computation principle. Nonetheless, the collected data of sensors participating in query processing remain private. After several rounds of query processing, the query result is determined and subsequently returned to users. By adopting secure multi-party computation, only the output of the query is public to the relevant paid users, while other private data are not. The analysis and evaluations indicate that the proposed protocol reliably computes query results, avoids the energy hole problem and is effective in terms of communication cost.

The main contributions of this paper include:

1) A novel privacy-preserving MAX/MIN query protocol is

proposed for WaaS, whereby two cooperative query processing algorithms are deployed in the aggregate sensor and normal sensors. To the best of our knowledge, this work is the first to discuss the privacy-preserving data query method in WaaS.

2) The query result's correctness, data privacy and network communication cost of the proposed protocol are analyzed. In addition, the solution to sensor failures in WSNs is presented, which is able to keep the proposed protocol serviceable.

3) Detailed quantitative experiments are performed to evaluate the query result's correctness and communication cost.

II. Related Work

Executing a data query is an important operation for event monitoring or data analysis in WSNs. Popular topics in data query research, such as privacy protection, integrity, and completeness verification, all involve security issues. Recently, secure range queries [3] [10] [26] [27] [30] [32], top-*k* query [9] [15] [16] [19] [20] [24] [31] and MAX/MIN queries [7] [8] [13] [25] [28] [29] have been broadly investigated. We mainly discuss the related work of the MAX/MIN query since it is the focus of our work.

In WSNs, Samanthula et al. proposed a secure MAX/MIN query method, SDAM [25], which adopts GM probabilistic encryption [12]. In the MAX/MIN computation, each bit of a data item collected by a sensor is encrypted into a vector of a length that equals the domain size of the collected data. The generated vectors are utilized to compute the MAX/MIN result by XOR-homomorphism. SDAM can protect data privacy even if a few sensors are compromised. However, the network communication cost could be very large if the domain size of the collected data is large. Yao et al. proposed a privacy-preserving MAX/MIN aggregation, called PMMA [28]. It adopts prefix membership verification (PMV) [4] [17] and HMAC to implement secure comparison without knowing the plaintext of data and then achieves privacy-preserving MAX/MIN aggregation. However, PMV and HMAC mechanisms greatly increase data transmission, which raises the network communication cost. Additionally, all sensors in PMMA share the same HMAC key, and data privacy could be breached if a sensor is compromised. Groat et al. proposed a non-cryptographic privacy-preserving MAX/MIN query method, KIPDA [13]. It obfuscates the sensitive query result by hiding it among a set of camouflage values, enabling *k*-indistinguishability for MAX/MIN aggregation. Since no cryptographic functions are used, KIPDA is efficient in query processing, and the network communication cost depends on the k-indistinguishable dataset size. However, this method's privacy protection is vulnerable if sensors are compromised and collude with each other. The risk of a leakage of private data rises as the number of colluding sensors increases.

Additionally, a few studies, such as PMQP [29], SMQ [7], and RSCS-PMQ [8], explore secure MAX/MIN queries in the two-tier wireless sensor network [11]. However, the schemes proposed in those studies cannot be utilized to implement privacy-preserving MAX/MIN queries for WaaS because those schemes were designed for different sensor network architectures.

According to the overview of the existing studies, all the above methods do not consider the subject of WSN-as-a-Service. In this paper, we will present a novel privacy-preserving MAX/MIN query protocol for WaaS.

III. Models and Problem Statement

A. WSN-as-a-Service Model

We assume that WSNs are deployed and maintained by owners who provide data queries as services to users. We name such model as WSN-as-a-Service (WaaS). There are n sensors in WSNs, $S = \{s_1, s_2, ..., s_n\}$. Since sensors have limited resources such as power, computation, storage, etc., they can only communicate with the neighboring sensors that are in their respective communication ranges. As a result, a multi-hop network is constructed. There is a proxy sensor $s_A \in S$, named as the aggregate sensor of the network, that acts as the bridge between users and the network. The responsibility of s_A is to broadcast queries from users to the entire network, cooperate with other sensors to complete queries and return query results to users. Theoretically, any sensor in the network could be selected as s_A, the reason for which will be discussed in Section $IV.E$.

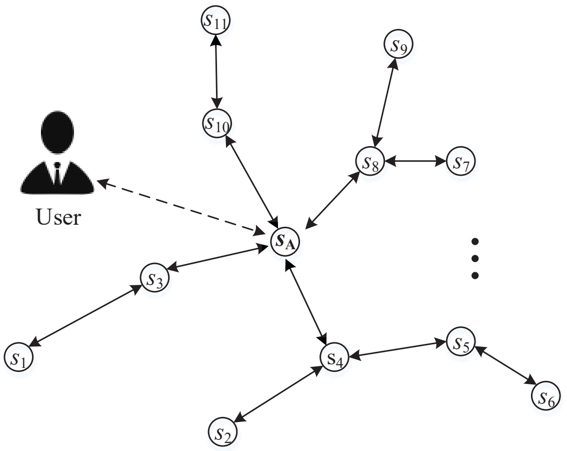

Fig. 1. An example of network topology based on TAG

A detailed description of WaaS is shown in Fig. 1, where the network is organized according to a tree routing topology based on the TAG protocol [22], and s_A is the root node. When a query's processing begins, sensors will perform data processing according to the established protocol and then upload the corresponding data to s_A hop by hop. Ultimately, s_A will calculate the query result and return it to users.

B. Query Model

A MAX/MIN query is a data request operation in WSNs used to obtain the maximum or minimum data from an area and time slot of interest, which can be formalized as a tuple with four elements: Q=(qid, S, t, MAX/MIN), where qid is the ID of the query, S is the set of queried sensors, t is the queried time slot, and MAX/MIN indicates the query type. For example, the objective of query Q=(100, $\{s_1, s_2, ..., s_{10}\}$, t, MAX) is to obtain the maximum from the collected data of sensors $\{s_1, s_2, ..., s_{10}\}$ at time slot t. For simplicity, we mainly discuss the MAX query; the MIN query's processing method is the same.

C. Problem Statement

In WaaS, the collected data of sensors in a WSN are owned by the network owners and are their private data. Users pay owners for query services and are authorized to access the result data of such paid queries. Therefore, query results are public to respective authorized users. However, users could be curious about attempting to obtain private data from WSNs. It is a challenge to protect private data from such curious users while simultaneously providing efficient data query services.

In this paper, the *honest-but-curious* threat model [2] is adopted. It assumes that a part of the sensors of WSNs could be compromised and collude with curious users to obtain private data from *innocent sensors* that are not compromised. However, the compromised sensors still strictly follow the established protocols. The scenario of the compromised sensors deliberately damaging or falsifying the collected data is outside the scope of this paper. In addition, it is reasonable for the network to have only a few compromised sensors. Otherwise, it will be unavailable.

The key issue of this paper is protecting the private data of innocent sensors from curious users. The main security objective of this paper is for each sensor $s_i \in S$, s_i only has knowledge of its collected data, and be unable to obtain any data from other sensors. Obviously, if we can achieve such an objective, even if a curious user has access to a few compromised sensors and makes them collude with the user, the user will nonetheless have no insight into any private data of other innocent sensors.

We know that sensors in WSNs are always equipped with batteries that have limited stored power, and the power of sensors is mainly consumed by communication between sensors; hence, the communication cost of WSNs is a critical metric of performance evaluation. We denote the network communication cost metric of a query processing as *E*. It is the sum of the communication cost of all sensors in the network as shown in Eq. (1):

$$E = \sum_{s_i \in S} E_i \tag{1}$$

where E_i is the communication cost of sensor s_i during query processing. In addition, avoidance of the energy hole problem [23] in the network, which is caused by excessive communication load of some critical nodes and could reduce the lifetime of the network, is another important metric.

D. Notations

The notations used in this paper are as follows.

- S — The set of n sensors in the network, where $S = \{s_1, s_2, ..., s_n\}$ and $id(s_i)$ is the ID of sensor s_i.
- s_A— The aggregate sensor of the network, $s_A \in S$, which is in charge of coordinating with users and processing queries.
- D_i — The data collected by sensor s_i during a time slot; such data are assumed to have β binary bits, i.e., $D_i = d_{i,1}d_{i,2}...d_{i,\beta}$, where $d_{i,j} \in \{0, 1\}$ and $j \in \{1, 2, ..., \beta\}$.
- $k_{i,j}$ — The *local cover code* of s_i for round j of processing, which is a generated key and is only known to s_i and network owners. We assume that $k_{i,j}$ has w bits, i.e., $k_{i,j} = \{0, 1\}^w$.
- K_j — The *global cover code* of s_A for round j of processing, which is formed according to Algorithm 1 and is only known to s_A.
- f_i — The flag indicating whether s_i is a candidate sensor that could contribute to the query result; $f_i \in \{0, 1\}$. If $f_i = 1$, then s_i is a candidate sensor; otherwise it is a non-candidate sensor.
- $P_{i,j}$ — The *primitive identification code* generated by s_i in round j of processing.
- $C_{i,j}$ — The *secure identification code* generated by s_i in round j of processing.
- r_j — The *cooperation request code* generated by s_A in round j of processing; $r_j \in \{0, 1\}$, which is also the jth bit of the query result.
- R — The result of a query.

In the above notations, local cover codes, global cover codes and secure identification codes are all secure codes that are utilized to implement privacy-preserving query processing. Additionally, $\{0\}^w$ and $\{0, 1\}^w$ are defined as a w-bit zero code and a w-bit nonzero code. Here, $\{0, 1\}^w$ represents a random w-bit code.

IV. Query Processing Protocol

In this section, we propose a privacy-preserving MAX/MIN query protocol (PMQ) for WaaS adopting the idea of secure multi-party computation. According to [6], we know that secure multi-party computation (SMC) is a subfield of cryptography with the goal of creating methods for parties to jointly compute a result over their inputs while keeping those inputs private. Thus, the objective of our proposed protocol is for the query result to be computed through cooperation of all sensors in the network and for the collected data of those sensors participate in the query result's computation to remain private.

We first describe the secure cover code generating algorithm and then present the details of privacy-preserving MAX/MIN query protocol that contains two cooperative query processing algorithms. Afterward, the analysis of query result's correctness, security and network communication cost is presented, and the solution to sensor failures is also proposed.

A. Cover Code Generation

Before the network is deployed, for each sensor $s_i \in S$, owners embed a private root key g_i and a key generator in s_i. Owners possess all the root keys of sensors. The key generator is denoted by $GenKey(qid, sid, num, rkey)$, where qid and

sid are respectively the IDs of a query and a sensor, num is the round number of a query, and $rkey$ is the root key of the sensor whose ID is sid. An example of $GenKey$ could be a hash-based key generator, such as $GenKey(qid, sid, num, rkey) = hash(qid||sid||num||rkey)$. Thus, the keys generated by $GenKey$ depend on the four parameters and are changed in different rounds of different queries.

When a MAX/MIN query service is started, owners generate global cover codes for the query and transmits them to s_A, while each sensor generates their own local cover codes for the query, respectively. The cover codes are the secure keys for privacy-preserving query processing and are generated as shown in Algorithm 1.

```
Algorithm 1: Generating Cover Codes
Input: $qid$, the root keys of sensors $\{g_1, g_2, ..., g_n\}$, and
       the number of rounds of a query processing $\beta$.
Output: The local cover codes of sensors,
        $\{k_{i,1}, k_{i,2}, ..., k_{i,\beta} \mid i \in \{1, 2, ..., n\}$; and the
        global cover codes of $s_A$, $\{K_1, K_2, ..., K_\beta\}$.
foreach $j \in \{1, 2, ..., \beta\}$ do
    foreach $s_i \in S$ do
        Owners and $s_i$ both generate the same local cover
        codes for $s_i$, $k_{i,j} = GenKey(qid, id(s_i), j, g_i)$;
    end
    Owners generate the global cover code for round $j$,
    $K_j = \oplus_{i=1}^{n} k_{i,j}$;
end
Owners transmit the $\beta$ global cover codes to $s_A$;
```

Algorithm 1 generates secure cover codes, such as local cover codes and global cover codes, for sensors and s_A. Each sensor except s_A has β respective local cover codes, while s_A has β local cover codes and β global cover codes. All those generated cover codes are nonzero codes. Since the cover codes are changed in different rounds of different queries and the root keys are owned by sensors privately, they will be used as the secure keys for privacy-preserving MAX/MIN queries.

B. Query Processing Protocol

We consider the MAX query as an example to discuss the protocol that is implemented by several rounds of interaction between sensors and s_A. Each round's processing follows the secure multi-party computation principle, which protects private data from curious users.

The flowchart of the protocol is shown in Fig. 2. During query processing, the secure cover codes are firstly generated and deployed according to Algorithm 1. The query command Q are broadcasts in the network and then s_A performs β rounds of interaction with other sensors. The proposed privacy-preserving MAX/MIN query protocol (PMQ) consists of two cooperative algorithms. One is the query processing algorithm in s_A, and the other is query processing in s_i. The details of the algorithms are described in Algorithm 2 and Algorithm 3.

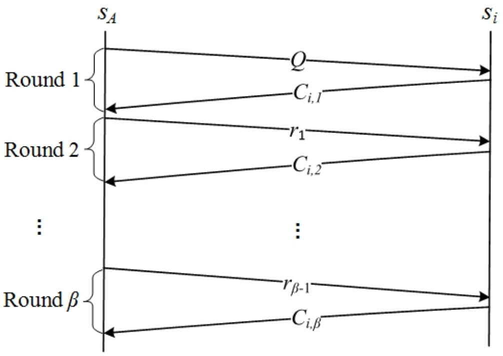

Fig. 2. The flow chart of the proposed query protocol

As shown in Algorithm 2 and 3, at the beginning of a query, the collected data $D_i = d_{i,1}d_{i,2}...d_{i,\beta}$ of s_i could be the query result; therefore, all sensors in the network are candidate sensors, and the corresponding flag f_i of each sensor s_i is initialized to 1. However, if f_i changes from 1 to 0, D_i must not be the query result. Once a sensor has been determined to be a non-candidate node, its primitive identification codes will always be zero codes in subsequent rounds. However, the corresponding secure identification codes are still private nonzero codes since they are generated by XORing local cover codes that are generated by a key generator. Obviously, after each round of processing, the number of candidate nodes decreases until it reaches only one node, and the query result is finally determined in s_A. Because each sensor has its own private local cover codes for each round of processing, the collected data of innocent sensors participating in query processing remain private individually.

Lemma 1. In Algorithm 2, the secure identification code computed by s_A in round j is $\oplus_{i=1}^{n} P_{i,j}$, i.e.,

$$C_{A,j} = \oplus_{i=1}^{n} P_{i,j}. \quad (7)$$

Proof. According to Algorithm 3, for any $s_i \in S - \{s_A\}$ in round j, s_i generates a code that is computed by XORing the result of $P_{i,j} \oplus k_{i,j}$ and the codes received from its child nodes (if it has any); afterward, s_i submits the generated code to its parent node. Such procedures are performed in sensors along the reversed tree routing paths from the leaf nodes to the root s_A. Finally, s_A computes $C_{A,j}$ according to Eq.(3). Thus, we have $C_{A,j} = ((P_{1,j} \oplus k_{1,j}) \oplus (P_{2,j} \oplus k_{2,j}) \oplus ... \oplus (P_{n,j} \oplus k_{n,j})) \oplus (\oplus_{i=1}^{n} k_{i,j}) = (\oplus_{i=1}^{n} P_{i,j}) \oplus (\oplus_{i=1}^{n} k_{i,j}) \oplus (\oplus_{i=1}^{n} k_{i,j}) = \oplus_{i=1}^{n} P_{i,j}$. Therefore, Lemma 1 holds. ■

According to Lemma 1, the secure identification code computed by s_A in round j is equal to the result of XORing the primitive identification codes in round j of all sensors in the network.

C. Correctness Analysis of Query Results

According to Algorithm 2 and Algorithm 3, many XOR operations need to be performed. The XOR operation of two nonzero codes could generate a zero code that could cause the query result to be incorrect. Hence, we present the correctness analysis of the query results in this section.

```
Algorithm 2: Query Processing in $s_A$
Input: $Q$ is the query command; $C_{A_1,j}, C_{A_2,j}, ..., C_{A_u,j}$
        are the secure identification codes received from
        $s_A$'s child nodes in round $j$;
        $D_A = d_{A,1}d_{A,2}...d_{A,\beta}$ is the collected data of $s_A$,
        where $d_{A,j} \in \{0,1\}$.
Output: The query result $R$
foreach round $j$ from $1$ to $\beta$ do
    if $j = 1$ then
        Broadcasts $Q$ in the network, and then waits for
        the responses (generated in Algorithm 3) from its
        child sensors in the first round;
    end
    After receiving its child nodes' secure identification
    codes of round $j$, $s_A$ compute its round $j$ primitive
    identification code:
        $P_{A,j} = \begin{cases} \{0\}^w & d_{A,j} = 0 \\ \{0,1\}^w & d_{A,j} = 1 \end{cases}$    (2)
    Compute its round $j$ secure identification code:
        $C_{A,j} = (\oplus_{x=1}^{u} C_{A_x,j}) \oplus P_{A,j} \oplus k_{A,j} \oplus K_j$    (3)
    Compute its round $j$ cooperation request code:
        $r_j = \begin{cases} 1 & C_{A,j} \neq \{0\}^w \\ 0 & C_{A,j} = \{0\}^w \end{cases}$    (4)
    if $j < \beta$ then
        Broadcasts $r_j$ in the network to start the next
        round processing and waits for its child nodes
        responses (generated in Algorithm 3) of the next
        round;
    else if $j = \beta$ then
        The query result has been computed where
        $R = r_1r_2...r_\beta$ and the query processing is
        finished;
    end
end
```

Lemma 2. We assume that $D_1, D_2, ..., D_m$ are w-bit nonzero codes where $m \geq 2$, i.e. $D_i \neq \{0\}^w$ and $i \in \{1,2,...,m\}$. The result of $\oplus_{i=1}^{m} D_i$ could turn to be $\{0\}^w$. The probability of $\oplus_{i=1}^{m} D_i = \{0\}^w$ is given by Eq.(8).

$$\Pr(\oplus_{i=1}^{m} D_i = \{0\}^w) = \sum\nolimits_{i=1}^{m-1} (-1)^{i-1} \cdot \frac{1}{(2^w-1)^i} \quad (8)$$

Proof. We use mathematical induction to prove this lemma.

(1) For $m = 2$, D_1 and D_2 could have $2^w - 1$ values since they are both w-bit nonzero codes. If and only if $D_1 = D_2$ holds, then $D_1 \oplus D_2 = \{0\}^w$, and its probability is given by Eq.(9).

$$\Pr(D_1 \oplus D_2 = \{0\}^w) = \frac{2^w - 1}{(2^w-1)\cdot(2^w-1)} = \frac{1}{2^w-1} \quad (9)$$

```
Algorithm 3: Query Processing in $s_i$
Input: $Q$ is the query command; $C_{i_1,j}, C_{i_2,j}, ..., C_{i_v,j}$
        are the secure identification codes received from
        $s_i$'s child nodes in round $j$; $r_{j-1}$ is the received
        cooperation request code in round $j-1$ ;
        $D_i = d_{i,1}d_{i,2}...d_{i,\beta}$ is the collected data of $s_i$
        where $d_{i,j} \in \{0,1\}$.
Output: The secure identification codes for each round
foreach round $j$ from $1$ to $\beta$ do
    if $j = 1$ then
        After $s_i$ receives the query command $Q$, it starts
        the first round processing;
        Set $f_i = 1$;
        Compute its round 1 primitive identification code:
            $P_{i,1} = \begin{cases} \{0\}^w & d_{i,1} = 0 \\ \{0,1\}^w & d_{i,1} = 1 \end{cases}$    (5)
    else
        After $s_i$ receives the cooperation request code
        $r_{j-1}$ broadcasted in the previous round, it starts
        the $j$th round;
        if $f_i = 0$ then
            $P_{i,j} = \{0\}^w$;
        else
            if $r_{j-1} = 1 \wedge d_{i,j-1} = 0$ then
                Set $f_i = 0$ and $P_{i,j} = \{0\}^w$;
            else
                Set  $P_{i,j} = \begin{cases} \{0\}^w & d_{i,j} = 0 \\ \{0,1\}^w & d_{i,j} = 1 \end{cases}$ ;
            end
        end
    end
    Compute its round $j$ secure identification code $C_{i,j}$
    according to its position in the routing tree:
        $C_{i,j} = \begin{cases} P_{i,j} \oplus k_{i,j} & s_i \ is \ a \ leaf \ node \\ (\oplus_{x=1}^{v} C_{i_x,j}) \oplus P_{i,j} \oplus k_{i,j} & otherwise \end{cases}$    (6)
    Submit $C_{i,j}$ to its parent node;
end
```

Therefore, Lemma 2 holds for $m = 2$.

(2) We assume that Lemma 2 holds for $m = k$, which indicates that $\Pr(\oplus_{i=1}^{k} D_i = \{0\}^w) = \sum\nolimits_{i=1}^{k-1} (-1)^{i-1} \cdot \frac{1}{(2^w-1)^i}$ holds. Then, we have

$$\Pr(\oplus_{i=1}^{k} D_i \neq \{0\}^w) = 1 - \sum\nolimits_{i=1}^{k-1} (-1)^{i-1} \cdot \frac{1}{(2^w-1)^i} \quad (10)$$

If $m = k+1$, because $D_{k+1} \neq \{0\}^w$, if and only if $\oplus_{i=1}^{k} D_i \neq \{0\}^w$ and $\oplus_{i=1}^{k} D_i = D_{k+1}$ both hold, then $\oplus_{i=1}^{k+1} D_i = (\oplus_{i=1}^{k} D_i) \oplus D_{k+1} = \{0\}^w$. Therefore, the probability of $\oplus_{i=1}^{k+1} D_i = \{0\}^w$ is shown as Eq. (11).

$$\begin{aligned}
&\Pr(\oplus_{i=1}^{k+1} D_i = \{0\}^w) \\
&\quad = \Pr(\oplus_{i=1}^{k} D_i \neq \{0\}^w) \cdot \frac{2^w - 1}{(2^w - 1)\cdot(2^w - 1)} \\
&\quad = \left(1 - \sum\nolimits_{i=1}^{k-1} (-1)^{i-1} \cdot \frac{1}{(2^w - 1)^i}\right) \cdot \frac{1}{2^w - 1} \qquad (11) \\
&\quad = \sum\nolimits_{i=1}^{k} (-1)^{i-1} \cdot \frac{1}{(2^w - 1)^i}
\end{aligned}$$

Hence, Lemma 2 holds for $m = k + 1$.

As a result, we conclude that Lemma 2 holds according to the above proof. ■

Definition 1. An encoding neutralization is the phenomenon of the result of XORing m nonzero codes turns out to be a zero code where $m \geq 2$.

For example, we assume that $\{x_1, x_2, ..., x_m\}$ are all nonzero codes where $m \geq 2$, but $\oplus_{i=1}^{m} x_i = \{0\}^w$ holds. Then, this is an instance of encoding neutralization.

Once an encoding neutralization occurs in a round of the query processing, the corresponding bit of the round in the query result turns from 1 to 0 and hence the query result is turned to be incorrect. We calculate the probability of getting correct query result in the following Lemma 3.

Lemma 3. Assuming that R is the result of query Q, if the encoding neutralizations happen during the query processing, then R is possibly correct with probability denoted by $\Pr(R)$. The calculation of $\Pr(R)$ is shown in Eq. (12).

$$\Pr(R) = \prod\nolimits_{i=1}^{\beta} (1 - Pr_{n_i}) \qquad (12)$$

where n_i is the number of sensors with primitive identification codes that are nonzero codes in round i, and Pr_{n_i} is the probability of encountering encoding neutralization in round i. According to Lemma 2, the calculation of Pr_{n_i} is shown in Eq. (13),

$$Pr_{n_i} = \begin{cases} \sum_{j=1}^{n_i - 1} (-1)^{j-1} \cdot \frac{1}{(2^w - 1)^j} & n_i \geq 2 \\ 0 & n_i = 1 \end{cases} \qquad (13)$$

Proof. Because the primitive identification codes of sensors are generated by a key generator with four input parameters, it is possible that encoding neutralizations occur when XORing those codes in some rounds, i.e., $\exists j \in \{1, 2, ..., \beta\}(\oplus_{v=1}^{n} P_{v,j} = \{0\}^w)$, where $P_{i,j}$ is the primitive identification code of s_i in round j. Once an encoding neutralization occurs in a round, the query result R will be incorrect. Thus, if and only if there is no encoding neutralization in all rounds, R is correct. As a result, the probability that R is correct is $\prod_{i=1}^{\beta} (1 - Pr_{n_i})$, where Pr_{n_i} is the probability of having encoding neutralization in round i, and the calculation is shown in Eq. (13). ■

In practice, it is possible that encoding neutralizations could occur in query processing. Therefore, the returned query result is correct with a certain probability. We will evaluate the probability in Section V.

D. Security Analysis

The proposed PMQ query protocol follows the idea of secure multi-party computation that can protect private data of innocent sensors from curious users even if a few sensors have been compromised and are in collusion with such users.

Lemma 4. The private data of innocent sensors can be protected from curious users even if there are compromised sensors in collusion with such users.

Proof. According to the algorithms of the PMQ protocol, curious users can only obtain the result of a query. Since the query processing procedures are performed cooperatively by sensors in WSNs, the way curious users successfully snoop on private data of innocent sensors is by compromising sensors and making them collude with curious users. However, the private data of innocent sensors remain protected. The analytical proof is as follows. We assume that s_i is an innocent sensor that has not been compromised by curious users, and data D_i is its private collected data. According to Algorithm 3, for each bit of D_i, s_i uploads the corresponding secure identification code to its parent sensor until the data reach s_A. The secure identification code is a secure code rather than the plaintext of D_i, which is generated by XOR operations as shown in Eq. (6), depending on the location of s_i in the routing tree and the round number. Since a local cover code is XORed in each secure identification code generation, and the local cover code is generated depending on a private root key of s_i and changed in different rounds of different queries, it is impossible for other sensors and curious users to exactly determine the corresponding bit of D_i reversely. Obviously, during multi-sensor cooperation in each round, a bit of the query result is determined, but the collected data involved in query processing remain private as a consequence of the principle of secure multi-party computation.

As a result, the private data collected by innocent sensors are well protected, and curious users cannot obtain the private data even if a few sensors are compromised and collude with such users. ■

E. Communication Cost Analysis

We assume that the length of a query command is l_q. The length of a cooperation request code, a secure identification code and a query result are 1, w and β, respectively. The network contains n sensors, including s_A. There are β rounds in each query's processing, which equals the length of the collected data. According to the algorithms in the protocol, the communication cost of the network consists of four parts:

- **Broadcasting query command**. Sensor s_A broadcasts the query command to other sensors in the network; thus, the communication cost of the query command broadcasting is $n \cdot l_q$ for a query.
- **Broadcasting cooperation request codes**. Sensor s_A broadcasts a cooperation request code to other sensors in each round except the last round of a query's processing; thus, the communication cost of cooperation request code broadcasting is $n \cdot (\beta - 1)$ in a query.

- **Uploading secure identification codes**. Each sensor except s_A uploads a secure identification code to its parent sensor in each round; thus, the communication cost of secure identification code uploading is $(n-1) \cdot \beta \cdot w$ for a query.
- **Returning query result**. Once s_A has obtained the query result, it returns the result to the authorized users; thus, the communication cost of returning the query result is β.

Therefore, we obtain the total communication cost of network E as Eq. (14).

$$E = n \cdot l_q + n \cdot (\beta - 1) + (n-1) \cdot \beta \cdot w + \beta \tag{14}$$

Additionally, in accordance with the above communication cost analysis, we observe that all sensors except s_A account for the same communication cost $l_q+(\beta-1)+\beta \cdot w$ during a query, while s_A accounts for communication cost $l_q + (\beta - 1) + \beta$, which is less than that of other sensors. Therefore, we draw two conclusions.

First, the communication cost of all sensors in the network is almost balanced, and the traditional energy hole problem caused by excessive communication load of some critical nodes is avoided. Thus, the lifetime of the network is maximized.

Second, the aggregate sensor s_A is not the bottleneck of the entire network in terms of communication cost since its communication cost is less than that of other sensors during a query. Thus, in theory, any sensor in the network can be selected as s_A. However, the choice of s_A depends on the practical requirements. For example, if users want to get the query result as soon as possible, a sensor in the central area of the network is better to be s_A. The reason is that the height of the routing tree is lower and less time is consumed on transmitting a message from the leaf nodes to s_A. But if the application environment is cumbersome, such as forest fire monitoring, s_A will most likely have to be at the edge of WSNs.

F. Solution to Sensor Failures

In WSNs, sensor failures could happen because of hardware or software error, etc., which could invalid the proposed PMQ protocol. For example, if sensor s_i has failed and does not participate in the query processing, then the secure identification code computed by s_A in round $j \in \{1, 2, ..., \beta\}$ of processing turns to be $C_{A,j} = (\oplus_{s_x \in S-\{s_j\}} P_{x,j}) \oplus k_{i,j}$. According to Eq. (4) in Algorithm 2, the determined jth bit of the query result, r_j, could be incorrect and thus the query result could be incorrect. To enhance the survivability of the PMQ protocol, it is necessary to design a solution to the problem caused by sensor failures.

In this section, we propose a countermeasure to survive the PMQ protocol encountering the sensor failures. Under conventional assumption, the normal sensors are always the majority while the failed sensors are few and occasional, otherwise, the network will be unstable and unavailable. Fault detection methods of WSNs [14] [18] can be adopted to detect failed sensors. Once there are failed sensors detected, the following strategies are performed.

- Assuming that the detected failed sensors are $S_F = \{s_1, s_2, ...s_t\}$, if s_A is in S_F, a normal sensor near the original s_A is chosen to be the new s_A.
- The tree route topology without those failed sensors is rebuilt according to the TAG protocol [22].
- The global cover code generation shown in the step 5 of Algorithm 1 is updated by Eq. (15).

$$K_j = \oplus_{s_i \in S-S_F} k_{i,j} \quad j \in \{1, 2, ..., \beta\} \tag{15}$$

After the above measures, the failed sensors are removed from the networks and the generation of global cover codes depending on the local cover codes of all normal sensors are refreshed. Thus, the invalidation to the PMQ protocol caused by the sensor failures is recovered and the aggregate sensor is able to obtain the correct query results again.

V. Evaluations

To evaluate the proposed PMQ protocol, we first measure the probability of the query result's correctness. Next, we evaluate the communication costs of the proposed PMQ protocol and three other related protocols, SDAM [25], PMMA [28] and KIPDA [13]. The former is measured according to the theoretical analysis in Section $IV.C$, while the latter is implemented on the improved simulator of [5] with the Intel Lab Dataset [21].

The evaluations are performed on a PC with an Intel i5-3230M 2.6GHz CPU and 8 G of RAM running Windows 7 OS, Eclipse, Java and MATLAB. The parameters used in the evaluations are the length β of the collected data, the length w of the secure identification code and the sensor number n with the default values of 16, 16 and 100, respectively. The default domain size of collected data in SDAM is the square root of the maximum of the default settings. The size of the set of camouflage values in KIPDA is one-half of the default number of sensors in the network.

A. Evaluation of Query Result's Correctness

We consider β, w and n as the independent variables to measure the probability $\Pr(R)$ of the query result's correctness. The evaluation results are shown in Fig. 3-5.

Fig. 3. shows that $\Pr(R)$ declines as β increases but remains above 99.96% for the given parameter settings of our experiments. Fig. 4. shows that, as w grows, $\Pr(R)$ increases dramatically and is almost 100% after $w = 2$. Fig. 5. indicates that the changes of n have little effect on $\Pr(R)$ and it is always above 99.98%.

According to the above measurement results, we observe that the probability of query result's correctness is very high and remains 99.9% for the given parameter settings of our experiments. This finding indicates that the proposed PMQ protocol has a very high accuracy of query processing.

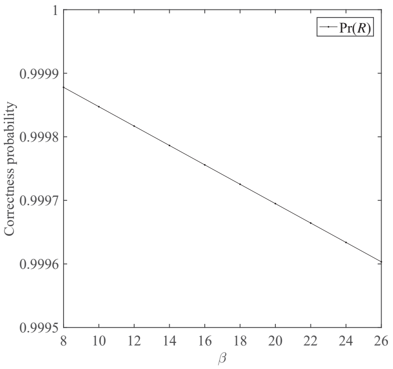

Fig. 3. $\Pr(R)$ versus β

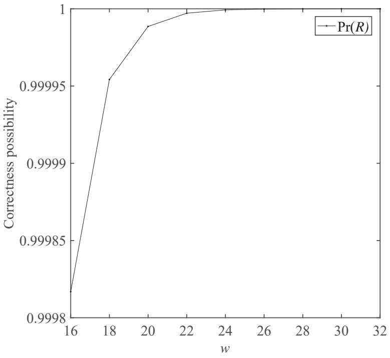

Fig. 4. $\Pr(R)$ versus w

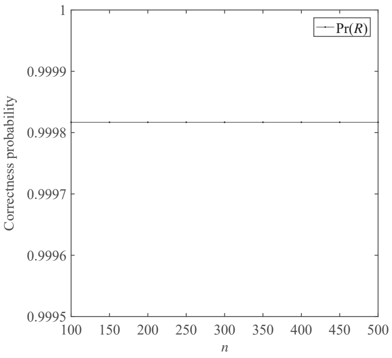

Fig. 5. $\Pr(R)$ versus n

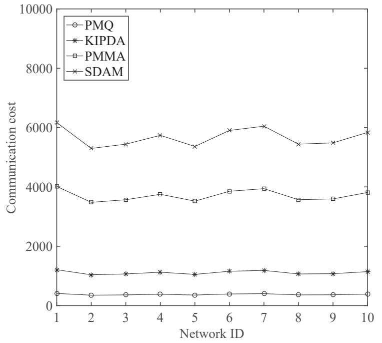

Fig. 6. Communication cost versus network IDs

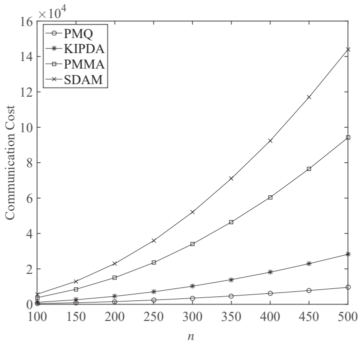

Fig. 7. Communication cost versus n

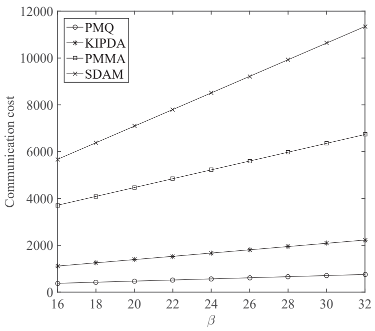

Fig. 8. Communication cost versus β

B. Evaluation of Communication Cost

In this evaluation, sensors are placed in a random distribution over 100×100 m^2 area, and the range of sensor communication is assumed to be 10 m. We assume that the packet transmissions are both collision-free and error-free in our experiments. In each measurement, we generate 10 different networks with different IDs. The experimental result is the average of those of 10 different networks. We consider network IDs, n and β as the independent variables to evaluate the communication cost of PMQ, KIPDA, SDAM and PMMA. The measurement results are shown in Fig. 6-8.

1) **Communication cost versus Network IDs**. As shown in Fig. 6., the communication costs of PMQ, KIPDA, PMMA and SDAM all change slightly and are distributed uniformly in different networks. The cost of SDAM is the highest, and that of PMQ is the lowest, while results of KIPDA and PMMA are between the former two. According to the statistical analysis, PMQ on average has approximately 93.36%, 89.85% and 66.12% lower communication cost than those of SDAM, PMMA and KIPDA on average respectively. The reason is that the size of transmitted data of each sensor in PMQ is much smaller than those of the other three methods.

2) **Communication cost versus** n. Fig. 7. indicates that the communication cost of PMQ, KIPDA, PMMA and SDAM all increase as n increases. It is obvious that the growth rate of the communication cost of SDAM is significantly higher than the growth rates of the other three methods, while PMQ achieves the lowest growth rate of communication cost. The reason is that the size of the total transmitted data packages in the four schemes all increase as the number of sensors rises. Furthermore, the bit-length of the transmitted data items in each sensor in PMQ is shorter than the respective lengths in KIPDA, PMMA and SDAM methods; thus, PMQ clearly has the lowest communication cost.

3) **Communication cost versus** β. Fig. 8. shows that as β increases, the communication costs of PMQ, KIPDA, PMMA and SDAM all increase. The reason is that the growth of β indicates that the bit-length of transmitted data items are increase in the four schemes. Additionally, their communication costs all increase correspondingly. Similar to the above results, PMQ has a lower communication cost than those of other methods.

As a result, we conclude that the proposed PMQ saves the communication cost greatly compared to those of SDAM, PMMA and KIPDA. It has a very high accuracy of query processing, and the probability of the query result being correct is higher than 99.9%.

VI. Conclusions

Data privacy protection is a popular topic in the research of WSNs and is urgently needed in many important fields, such as healthcare, national defense, etc. With the widespread adoption of WSNs, WSN-as-a-Service (WaaS) becomes economical and practical in applications where owners of WSNs provide data queries as services, while users pay for needed services as they use such services. In this article, which is the first to discuss a privacy-preserving data query method in the WaaS environment, we propose a privacy-preserving MAX/MIN query protocol for WaaS. The proposed protocol adopts the idea of secure multi-party computation that consists of two cooperative query processing algorithms. During query processing, multiple rounds of secure interactions between sensors are performed to compute the query result, while the collected data of sensors participating in query processing remain private. The presented analysis and evaluations show that the proposed protocol performs reliably, preserves privacy while computing query result, avoids the energy hole problem and is efficient in terms of network communication cost.

Acknowledgment

This research was supported by the National Natural Science Foundation of China under the grant Nos. 61872197, 61572263, 61672297 and 61872193; the Natural Science Foundation of Jiangsu Province under the grant No. BK20161516; the Natural Research Foundation of Nanjing University of Posts and Telecommunications under the grand No.NY217119.

References

[1] M. Amardeep, B. Rami, S. Fredrik, G. Harald, and E. Erik. Calvin constrained - a framework for iot applications in heterogeneous environments. In *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems*, pages 1063–1073. IEEE, June 2017.

[2] V. Bozovic, D. Socek, R. Steinwandt, and V. I. Villányi. Multi-authority attribute-based encryption with honest-but-curious central authority. *International Journal of Computer Mathematics*, 89(3):268–283, 2012.

[3] F. Chen and A. X. Liu. Privacy- and integrity-preserving range queries in sensor networks. *IEEE/ACM Transaction on Networks*, 20(6):1774–1787, 2012.

[4] J. Cheng, H. Yang, S. H. Y. Wong, P. Zerfos, and S. Lu. Design and implementation of cross-domain cooperative firewall. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP2007)*, pages 284–293. IEEE, October 2007.

[5] A. Coman, M. A. Nascimento, and J. Sander. A framework for spatio-temporal query processing over wireless sensor networks. In *Proceedings of the 1st Workshop on Data Management for Sensor Networks, in conjunction with VLDB*, pages 104–110. ACM, August 2004.

[6] R. Cramer, I. Damgård, and U. M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2000)*, pages 316–334. Springer, May 2000.

[7] H. Dai, M. Wang, X. Yi, G. Yang, and J. Bao. Secure max/min queries in two-tiered wireless sensor networks. *IEEE Access*, 5:14478–14489, 2017.

[8] H. Dai, T. Wei, Y. Huang, J. Xu, and G. Yang. Random secure comparator selection based privacy-preserving MAX/MIN query processing in two-tiered sensor networks. *Journal of Sensors*, 2016:1–13, 2016.

[9] H. Dai, G. Yang, H. Huang, and F. Xiao. Efficient verifiable top-k queries in two-tiered wireless sensor networks. *KSII Transactions on Internet & Information Systems*, 9(6):2111–2131, 2015.

[10] L. Dong, X. Chen, J. Zhu, H. Chen, K. Wang, and C. Li. A secure collusion-aware and probability-aware range query processing in tiered sensor networks. In *Proceedings of the 34th IEEE Symposium on Reliable Distributed Systems (SRDS 2015)*, pages 110–119. IEEE, September 2015.

[11] O. Gnawali, K. Jang, J. Paek, M. A. M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler. The tenet architecture for tiered sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys 2006)*, pages 153–166. ACM, November 2006.

[12] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 228(2):270–299, 2014.

[13] M. M. Groat, W. He, and S. Forrest. Kipda: k-indistinguishable privacy-preserving data aggregation in wireless sensor networks. In *Proceedings of 30th IEEE International Conference on Computer Communications (INFOCOM 2011)*, pages 2024–2032. IEEE, April 2013.

[14] M. Lee and Y. Choi. Fault detection of wireless sensor networks. *Computer Communications*, 31(14):3469–3475, 2008.

[15] R. Li, A. X. Liu, S. Xiao, H. Xu, B. Bruhadeshwar, and A. L. Wang. Privacy and integrity preserving top-k query processing for two-tiered sensor networks. *IEEE/ACM Transactions on Networking*, 25(4):2334–2346, 2017.

[16] J. Liang, C. Jiang, X. Ma, G. Wang, and X. Kui. Secure data aggregation for top-*k* queries in tiered wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 32(1-2):51–78, 2016.

[17] A. X. Liu and F. Chen. Collaborative enforcement of firewall policies in virtual private networks. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing*, pages 95–104. ACM, August 2008.

[18] X. Luo, M. Dong, and Y. Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers*, 55(1):58–70, 2006.

[19] X. Ma, X. Liu, J. Liang, Y. Li, R. Li, W. Ma, and C. Qi. A comparative study on two typical schemes for securing spatial-temporal top-*k* queries in two-tiered mobile wireless sensor networks. *Sensors*, 18(3):871, 2018.

[20] X. Ma, H. Song, J. Wang, J. Gao, and G. Min. A novel verification scheme for fine-grained top-k queries in two-tiered sensor networks. *Wireless Personal Communications*, 75(3):1809–1826, 2014.

[21] S. Madden. Intel lab data. http://db.csail.mit.edu/labdata/labdata.html. Accessed June 2, 2004.

[22] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. In *Proceedings of the 27th Annual ACM Symposium on Principles of Distributed Computing*, pages 95–104. ACM, December 2002.

[23] R. E. Mohemed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra. Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks. *Computer Networks*, 114:51–66, 2017.

[24] H. Peng, X. Zhang, H. Chen, Y. Wu, Y. Wu, and J. Zeng. Enable privacy preservation and result verification for top-k query in two-tiered sensor networks. In *Proceedings of the 2015 IEEE TrustCom/BigDataSE/ISPA*, pages 555–562. IEEE, August 2015.

[25] B. K. Samanthula, W. Jiang, and S. Madria. A probabilistic encryption based MIN/MAX computation in wireless sensor networks. *Wireless Personal Communications*, 75(3):1809–1826, 2014.

[26] J. Shi, R. Zhang, and Y. Zhang. A spatiotemporal approach for secure range queries in tiered sensor networks. *IEEE Transactions on Wireless Communications*, 10(1):264–273, 2011.

[27] Y. Tsou, C. Lu, and S. Kuo. SER: secure and efficient retrieval for anonymous range query in wireless sensor networks. *Computer Communications*, 108(4):1–16, 2017.

[28] Y. Yao, L. Ma, and J. Liu. Privacy-preserving max/min aggregation in wireless sensor networks. *Advances in Information Sciences and Service Sciences*, 4(6):272–295, 2012.

[29] Y. Yao, L. Ma, and J. Liu. Privacy-preserving max/min query in two-tiered wireless sensor networks. *Computers and Mathematics with Applications*, 65(9):1308–1325, 2013.

[30] Y. Yi, R. Li, F. Chen, A. X. Liu, and Y. Lin. A digital watermarking approach to secure and precise range query processing in sensor networks. In *Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM 2013)*, pages 1950–1958. IEEE, April 2013.

[31] C. Yu, G. Ni, I. Chen, E. Gelenbe, and S. Kuo. Top-k query result completeness verification in tiered sensor networks. *IEEE Transactions on Information Forensics and Security*, 9(1):109–124, 2014.

[32] J. Zeng, L. Dong, Y. Wu, H. Chen, C. Li, and S. Wang. Privacy-preserving and multi-dimensional range query in two-tiered wireless sensor networks. In *Proceedings of the 2017 IEEE Global Communications Conference*, pages 1–7. IEEE, December 2017.