PPRQ: Privacy-Preserving MAX/MIN Range Queries in IoT Networks

Savio Sciancalepore, Roberto Di Pietro
Division of Information and Computing Technology
College of Science and Engineering, Hamad Bin Khalifa University - Doha, Qatar
{ssciancalepore, rdipietro}@hbku.edu.qa

***Abstract*— Range queries are widely used in several Internet of Things (IoT) applications as a general strategy to improve the efficiency of the system. However, the communication patterns generated by the IoT nodes could lead to the identification of the devices satisfying the query, as well as to the disclosure of the queried data. State-of-the-art solutions to address the cited security issues rely on dedicated edge/fog nodes, whose deployment could be too expensive or challenging, especially in unattended scenarios where the installation of ad-hoc locations could be difficult and mains-supply is hardly available.**

In this paper, we propose PPRQ, a resilient, scalable, and lightweight protocol that allows privacy-preserving range queries in IoT networks. PPRQ is a probabilistic scheme that can be easily adapted to MIN, MAX, and MAX/MIN range queries, while requiring only hashing and bitwise xor operations. We show that PPRQ is robust, as it can be configured to provide over 99.9% accuracy in the query results. We also prove its resiliency against passive and active adversaries for a number of interesting and realistic scenarios. Our results are rooted in sound probability theory and supported by an extensive simulation campaign, while comparisons against state-of-the-art solutions show the flexibility and adaptability of PPRQ, especially for remote and unattended scenarios. Finally, further research directions opened up by the proposed solution are also highlighted.

***Index Terms*—IoT, Range Queries, Security, Privacy, Resilience.**

I. Introduction

Internet of Things (IoT) networks nowadays are pervasively diffused in home, office, and outdoor environments, and they seamlessly provide enhanced services in many application scenarios, such as buildings management, factory automation, critical infrastructures, transportation, agriculture, and military operations, to name a few [1], [2].

Many applications of modern IoT networks often require the computation of aggregated statistics, such as the Maximum (MAX) and Minimum (MIN) value sensed by the leaf IoT devices (namely, *MAX/MIN queries*, as well as the query of MAX/MIN values within pre-defined range values, e.g., e.g., MAX/MIN values in the interval $[a, b]$, with $a, b \in \mathbb{R}$. These queries are referred to as *MAX/MIN range queries* [3] [4].

Performing MAX/MIN range queries on large and distributed IoT networks could be useful in a variety of application scenarios. In *Smart Buildings*, it could be useful to obtain information about the maximum density of people in a room, to tune appropriately the intensity of the air conditioning and the luminosity of the lights, to optimize energy consumption [5]. In *Smart Oil and Gas Extraction Fields*, the operator could be interested in obtaining range data about the density of underground materials, to know if a specific material is present in a given underground area [6]. Within *Smart Cities*, it could be useful to obtain data about the maximum level of sensed $PM10$ emissions (Particulate Matter polluters), to promptly emit an alarm when a threshold is exceeded. Similarly, in areas closed to an airport, it could be useful to obtain the maximum level of sensed noise, to realize if the departure routes of the airplanes are appropriate or should be moved to other locations [7]. In *Smart Agriculture* applications, the operator could be interested in obtaining data about the maximum sensed humidity in a given field, to decide when to turn on artificial irrigation [8]. Finally, in *Military Applications*, when IoT sensors are deployed in hostile fields, it could be useful to know if the sensed number of enemy forces is within a given range, to prepare and deploy appropriate countermeasures [9].

All the above-mentioned applications require both *query privacy* and *location privacy*. Specifically, *Query Privacy* refers to the protection of the queried data, originated by the IoT sensors, from both passive and active attacks launched by adversaries deployed in the field. Indeed, the data acquired from the IoT devices should be appropriately protected while they travel throughout the wireless network up to the *sink IoT node*, directly connected to the system administrator. Conversely, we refer to *Location Privacy* as the protection of the location and identity of the node generating the MAX/MIN queried value. Indeed, assuming the *sink IoT node* can be compromised by a powerful adversary, the leakage of information in its possession should not directly reveal the identity and location of the leaf IoT device originating the data. This is even more important when considering the recent trend of *participatory* and/or *crowdsourced* IoT networks, where external devices can join and leave the network dynamically, while still providing valuable data [10] [11].

A few contributions in the last years proposed solutions to achieve privacy-preserving range queries in IoT networks (see Section II for an overview). However, almost all the proposals rely on powerful routers and intermediate nodes, realizing network architectures compliant with the emerging Edge and Fog Computing paradigms [12], [13]. We observe

that such an architectural design forces the operators to deploy powerful edge/fog nodes in dedicated locations, where a power source is available. This could be not always possible, due to limitations of the operating scenario—e.g., lack of mains-supply available in place, or lack of trusted deploying locations. Moreover, where feasible, compared with traditional IoT network architectures, deploying edge/fog nodes increases the cost of the solution, as these network elements should be fully-fledged computing elements.

Overall, the following technical challenges motivate our contribution:

- Designing a privacy-preserving MAX/MIN range query protocol that provides both *query privacy* and *location privacy*;
- The designed protocol should not require high computational or storage effort on intermediate aggregator devices, thus not forcing the deployment of dedicated *edge/fog* nodes;
- The designed protocol should not require the support for complex cryptography operations on IoT sensors;
- The designed protocol should reduce at minimum the energy consumption on IoT sensors.

Contribution. In this paper, we propose PPRQ, a probabilistic, lightweight, tunable, scalable, low-cost, and privacy-preserving protocol for MAX/MIN range queries in large IoT networks. PPRQ protects both the privacy of the queried data and the location privacy of the IoT node(s) originating the data. PPRQ trades off a slight increase in the communication overhead with a reduced error probability. For instance, we can achieve less than 0.01% of error probability, by requiring only a slight increase of the bandwidth overhead, independently from the number of IoT nodes in the networks. Furthermore, PPRQ is extremely robust against active adversaries, trying to corrupt the protocol computation. Indeed, compromising any fraction (less than the total) of these nodes would not lead to any significant advantage. The same result applies when it comes to the privacy of the computation.

Compared to the current literature on privacy-preserving range queries, PPRQ achieves query privacy and location privacy, without requiring neither the deployment of costly edge/fog nodes nor support for cryptography functions on the IoT nodes. These features make PPRQ much more flexible than competing solutions, as it enables privacy-preserving MAX/MIN range queries in a variety of scenarios, where the system administrator could not have the possibility to rely on dedicated mains-supplied locations, and/or the sensor nodes could not afford cryptography operations.

Specifically, the contributions provided by PPRQ are the following:

- PPRQ is a lightweight and low-cost solution for MAX/MIN range queries in large IoT networks, as it requires only a single hash and a single bitwise xor operation on both leaf IoT devices and aggregators;
- the accuracy of PPRQ is tunable, as its parameters can be configured in a way to trade-off a slight increase in the communication overhead with a reduced error probability. For instance, PPRQ enjoys a less than 0.01% of error probability using a replication factor $K \geq 12$;
- PPRQ preserves the privacy of both the result of the MAX/MIN query and the identity and location of the node originating the queried value;
- PPRQ is also robust against active adversaries, as compromising any fraction (less than the total) of the leaf IoT devices would not lead to any significant advantage for the adversaries. The same result applies when it comes to the privacy of the computation;
- compared to the current literature on MAX/MIN range queries in IoT networks, PPRQ emerges as the only solution that can guarantee both query privacy and location privacy, without requiring demanding cryptography operations on leaf IoT devices and high computational/storage efforts on the aggregator nodes; and,
- compared to solutions characterized by a low computational effort on the aggregator IoT devices, PPRQ emerges also as an energy-friendly protocol, requiring at least 33% less energy than the most energy-aware competing approach.

We remark that all the above features are useful in several use-cases, e.g., when the IoT sensors are very constrained and cannot support any cryptographic techniques, or when the deployment of edge/fog nodes is not feasible. To the best of our knowledge, PPRQ is the first protocol that allows to preserve *query privacy*, *location privacy*, and *query accuracy* by considering all these constraints on the participating devices.

Roadmap. The rest of this paper is organized as follows: Section II reviews the recent related work on privacy-preserving range query; Section III illustrates the scenario tackled in this work and the assumed adversarial model; Section IV provides the details of PPRQ; Section V includes important security considerations; Section VI provides the performance of PPRQ, as well as qualitative and quantitative comparisons with the current literature; and, finally, Section VII concludes the paper.

II. Related Work

Several solutions in the literature have been proposed to achieve privacy-preserving range queries. The privacy features of the different schemes can refer either to the specific value queried by the network administrator or to the identity and location of the node(s) providing the queried values. In Section II-A we describe the contributions dealing explicitly with privacy-preserving range queries, while Section II-B reports other contributions that focus on the protection of the location of the source of the message.

A. Privacy-Preserving Data Querying

The authors in [14] conceived a couple of protocols to guarantee privacy-preserving maximum and minimum computation in a Wireless Sensor Network (WSN). The protocols allow the computation of the max/min value and the identification of the node computing such value. However, if an attacker compromises the base station, it can get immediately the information on the specific node sensing the max (min).

In addition, the maximum (minimum) value computed by the protocol can be obtained both by a passive and active eavesdropper. Indeed, the protocol works bit-by-bit and sends out a broadcast notification to all the nodes in the network only when a bit 1 is computed. Therefore, the adversary can monitor the channels of the base station and understand which value has been computed.

Recently, along the same line of research, the authors in [15] proposed a query protocol leveraging secure multiparty computation. The protocol is based on two query processing algorithms, that are run both in aggregators and regular leaf IoT devices. The protocol involves multiple rounds of secure interactions between sensors and, in each round, one bit of the query result is determined in a privacy-preserving fashion. The protocol protects the result of the query, but not the identity of the generating leaf IoT sensors. Another non-cryptographic scheme to achieve privacy-preserving MAX/MIN range queries in IoT networks was provided by the authors in [16]. Specifically, the authors used a non-cryptographic method which obfuscates data by adding a set of camouflage values into the packets, achieving a form of k-indistinguishability. We notice that this protocol provides the privacy of the query result, but it does not protect the identity and location of the device(s) generating the query. The authors in [17] proposed the SafeQ protocol, preventing any adversary from obtaining information about both the query issued by the sink node and the data collected by the devices. The protocol can also detect compromised nodes in case of explicit misbehaving, and it preserves privacy by allowing to process query over encoded data, hiding their real value. The scheme can be also optimized using bloom filters, thus reducing the communication cost on the sensor devices. However, being based on the use of bloom filters, the scheme can incur in false-positives, that can be mitigated only by increasing the computational and bandwidth overhead of the solution. In addition, the scheme relies on *prefix membership verification* operations performed on intermediate storage nodes, requiring these nodes to be powerful and mains-supplied. In [18] is presented an efficient solution to private set intersection that, unlike competing proposals, does not resort to asymmetric crypto primitives and comes with provable (probabilistic) guarantees on the returned results. The authors in [19] proposed a privacy-preserving protocol for maximum and minimum queries, that prevents adversaries from gaining sensitive information from sensor collected data. To improve the privacy of the sensed data, the protocol leverages Prefix Membership Verification, enabling the verification of the maximum and the minimum without leaking any information on the actual value. Similarly to the previous work, prefix membership verification is performed on intermediate nodes, requiring them to be powerful and mains-supplied. In the context of vehicular networks, the authors in [20] proposed a secure querying scheme, where the roadside units (RSUs) act as fog devices, caching data closer to the vehicles and disseminating them at the time of the query. The scheme uses invertible matrices and aggregates ciphertext data using the homomorphic Paillier crypto-system, resulting in high overhead even on the sensing devices. Homomorphic encryption strategies are also used in [21], where the authors present a protocol that protects both the query range and individual IoT device's data.

While the previous schemes focused strictly on the computation of the MAX/MIN, in the literature it is possible to find also works focusing on top-K queries. For instance, the authors in [22] propose a top-k query processing scheme, able to achieve the integrity of the result of the query and the privacy of the sensed information. To improve the privacy level, the authors used pseudo-random hash functions and Bloom filters, and they translate top-k queries in top-range ones. However, their scheme suffers from false-positives issues due to the integration of the bloom filters, which can be mitigated only by increasing (largely) the bandwidth overhead. Besides, the time to obtain the result of the query increases linearly with the size of the network.

The authors in [23] proposed an architecture for secure querying IoT networks based on fog computing. Starting from a thorough analysis of the related schemes, they found that cloud queries on large IoT networks usually require time- and bandwidth-consuming interactions involving IoT devices. Therefore, they propose the adoption of dedicated fog nodes to mitigate the issue. However, in line with the previous protocols, fog nodes are assumed to be fully-fledged laptops, with reduced constraints on energy availability and computational requirements. We also mention the work by the authors in [24], using generalized distance-based techniques, modular arithmetic range query mechanisms, and circular modular verification schemes to achieve privacy- and integrity-preserving range queries. However, the scheme does not preserve location-privacy, as the sink IoT node can recover the identity of the sensor sensing the queried value. The authors in [25] presented some collusion-aware privacy-preserving range query protocols (CPRQ) based on classical privacy-preserving range query protocols. Specifically, considering the possibility that nodes collude to obtain the value queried by the sink node, they design an ad-hoc encoding scheme, to preserve the privacy of data and queries, and a verification strategy to verify the integrity of the result. However, their solution requires a non-negligible computational burden on intermediate master nodes, that need to be fully-fledged mains-supplied computational units. The authors in [26] addressed the problem of malicious aggregators, by proposing a protocol that is based on a bucketing technique, used to mix the data in a certain range, and a verifiable query protocol which employs encoding numbers to enable the sink to validate the reply. The protocol has been further extended in [27] to provide contextual event detection. Secure MAX/MIN range queries were also investigated by the authors in [28], proposing a scheme based on secure comparators and HMAC functions. HMACs are also used for MAX/MIN range queries by the authors in [29], in conjunction with symmetric encryption algorithms. However, both schemes are characterized by a significant communication overhead, which is usually a severe challenge in IoT networks. Considerations similar to the previous works apply also for the recent contribution in [30], where the IoT sensors have to support cryptography operations, while the aggregators are realized through expensive fog nodes.

Within the context of smart grids, many contributions such

as [31] and [32] proposed solutions for privacy-preserving range queries. Specifically, the authors in [31] proposed PaRQ, a scheme that works over encrypted metering data, to address the privacy issues in financial auditing for smart grids. Users store encrypted data on the cloud, while an authorized requester can send range query tokens to the cloud server to retrieve the metering data. When applied in the context of our work, PaRQ would require cryptography operations and communication overhead on the IoT sensors, thus being unsuitable for low-end devices. Similarly, the authors in [32] focused on protecting the communication link between the sensors of the smart grid, and on protecting the privacy of the data of the participating users. However, being the context of this referenced work different from the one tackled in this contribution, an adversary accessing the system would be able to identify the location of the IoT sensors providing the queried values.

To sum up, we notice that all the above schemes introduce a non-negligible computational burden on intermediate routing devices. While this architectural choice can be motivated through the adoption of the novel edge and fog computing architectures, it has drawbacks in terms of energy consumption and constraints required on the scenario. Indeed, the deployment of edge/fog nodes requires a dedicated location, under the control of the system administrator, where mains-supply or any other form of constant energy source is available. In addition, when mains-supply is not an issue, this setup increases the overall cost of the solution.

Finally, we highlight that privacy-preserving range queries were also widely investigated in the context of Cloud-based databases. For instance, the authors in [33] proposed a range query scheme satisfying the constraints of node and structure indistinguishability, characterized by a linear complexity in the number of data items in the query result. In addition, the authors in [34] tackled the problem of privacy-preserving conjunctive queries processing, where the queries contain both keyword conditions and range conditions on public clouds, by proposing an Indistinguishable Bloom Filter (IBF) data structure for indexing.
Overall, the above contributions provide very important results in the context of Cloud-based databases. However, they tackle a different scenario than the one considered in this paper. Indeed, privacy-preserving range query protocols suitable for IoT networks are characterized by different and more stringent constraints, such as the limited computational capabilities and the reduced energy availability of the nodes. Therefore, these constraints make the previous techniques hardly applicable in the context of privacy-preserving range queries in IoT networks.

B. Preserving Source Location Privacy via Anonymization

Many approaches in the literature have been proposed to protect the location of the source of a message [35], [36], [37]. The easiest way to achieve this objective is to anonymize the identity of the sender. For instance, the authors in [38] proposed a novel dummy location privacy-preserving (DLP) algorithm, taking into account both computational and privacy requirements of different users. Specifically, the proposed scheme allows selecting dummy locations in a greedy manner, reaching a trade-off between computational cost and privacy.

The authors in [39] introduced location labels, to differentiate locations of closely-related users. They designed a location-label based (LLB) algorithm for protecting location privacy, able to minimize the response time from location-based services. They also design a scheme to exchange pseudo-identifiers, to protect the identity of contributing users.

In the context of Industrial Internet of Things (IIoT) networks, the authors in [40] proposed a method for protecting location privacy based on differential privacy, maximizing, at the same time, the utility of the data. They combine the utility with privacy features, building an information tree model where the *index* mechanism included in the differential privacy is used to optimize the selection of the data, according to the frequency in the access of the data. To add noise to the data, they used the Laplace scheme, masking the frequencies in data access.

These schemes can be enriched with privacy-preserving features. One option could be to combine each of the above schemes with one of the techniques described in Subsection II-A, inheriting their strengths and weaknesses. Another (simplest) option could consist of combining each of the above strategies with standard symmetric/asymmetric cryptography techniques, to achieve both source location privacy and privacy-preserving range queries. However, this would require the support for encryption schemes on the IoT sensors originating the data, that could be not always available for low-end IoT sensors. In addition, such strategies could hide the location of the originating node when the data flows throughout the network, but the sink IoT node would know the identity of the specific IoT sensor originating the data. This constitutes a potential privacy threat, especially if the sink IoT node is compromised by the adversary.

III. Scenario and Adversary Model

In this section, we describe the system and the adversary models assumed throughout this work. Specifically, Section III-A focuses on the description of the scenario and system model, while the adversarial model is described in Section III-B.

A. System Model

In this paper, we assume a generic IoT network, as shown in Figure 1, where N nodes, namely $(s_1, s_2, \ldots, s_i, \ldots, s_N)$, can sense the surrounding environment. We assume three different types of nodes take part in the network: (i) IoT sensors, (ii) aggregators, and (iii) the sink IoT node.

The *IoT sensors* are low-end constrained IoT devices, and they are equipped with sensing and (limited) computational capabilities. Specifically, we assume that the sensors can collect several values from the surrounding environment, such as temperature, humidity, light, acceleration, magnetic field, and CO^2 level, to name a few. From the cryptography perspective, all we need is the capability to compute a hash. Despite nowadays IoT devices are becoming even more powerful, often

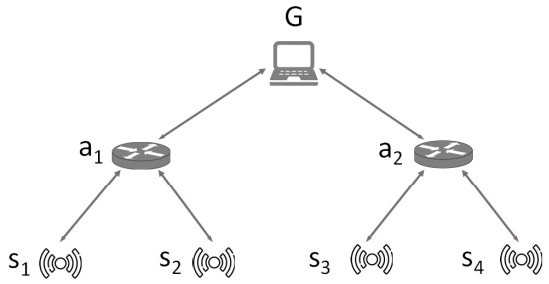

Fig. 1. Reference Scenario. Without loss of generality, we assume an IoT network arranged in a binary-tree topology, with multiple sensors (S), aggregators (A), and a sink IoT node (G).

the support for cryptography algorithms requires a dedicated crypto-processor, not always available at low-cost. Conversely, hashing functions can be executed efficiently via software, not requiring any specific hardware support [41].

The *aggregators* are intermediate nodes, whose aim is to receive data from the IoT sensors in direct RF visibility, aggregate them, and forward the result of the aggregation to either other aggregators or the sink IoT node. From the computational perspective, aggregators are like sensors, providing hashing functionalities. At the same time, we assume they can include dedicated hardware techniques to protect them against cloning, tampering, and physical compromise [42].

Finally, the *sink IoT node* is the gateway of the IoT network. Therefore, we assume it provides the capabilities of the aggregators as well as a link towards the public Internet—this latter one used to receive commands and reply with the requested data. In particular, in this paper, we assume that the queries issued by the network administrator through the sink IoT are of the following types: *MAX*, *MIN*, and MAX/MIN *range* queries.

We assume that each node, being it a sensor or an aggregator, is equipped with a value k, which can be a random value of the desired size. This value is shared between a single node and the upper-layer aggregator IoT node. For instance, with reference to the topology shown in Figure 1, the random value in possession of the node s_1, namely k_{s_1}, is shared between s_1 and a_1. In turn, a_1 shares a random value k_{a_1} with the sink IoT node G. We anticipate that this random value, despite being similar to an encryption key, is not used within symmetric or asymmetric encryption algorithms (see Section IV for more details). The specific means used by the IoT sensors to obtain the random value k is out of the scope of our paper. However, we highlight that such values can be either pre-configured by the network administrator in the involved devices, or obtained by the directly-connected devices dynamically, by resorting to non-cryptographic key agreement schemes working at the MAC layer, such as [43] and [44]. Moreover, the IoT sensors can resort to schemes based on Physical Unclonable Function (PUF) to authenticate their messages, such as the ones described in [45] and [46].

Note that, contrary to the majority of the schemes in the literature, we do not assume any time synchronization between the nodes involved in the scenario. Indeed, each device can refer to its local clock, and simply reply to the queries issued by the sink IoT node according to the protocol, as defined in Section IV. If an aggregator does not receive a message from one of the devices it is directly connected with, the aggregator will assume that such devices sent a value 0, and will proceed further with the protocol. Why the value 0 and its implication for the protocol will be explained later on.

In the following, to ease the discussion, we assume that the query issued by the network administrator through the sink IoT node is a *MAX* query, where the sink IoT node asks for the maximum value sensed by the IoT devices for a particular metric. However, we notice that other queries can be obtained as special cases of the *MAX* query. For instance, the *MIN* query can be easily obtained by flipping all the bits of the value sensed by each IoT sensor, while *MAX/MIN range queries* can be achieved by requiring the IoT sensors to set to the value 0 all the readings that are not included in the specific range query, e.g., all the values not included in the range $[a, b]$. Besides, as mentioned by the authors in [17], any query performed in a generic sensor network can be modeled as a range query, e.g., applying a dichotomic search algorithm.

Finally, in the rest of this discussion, we assume a binary tree logical topology, as shown in Figure 1. However, the protocol proposed hereby does not need any specific network topology, being adaptable to any physical or logical topology.

We report in Table I the main notation used throughout this paper. Note that boldface lowercase letters, e.g., $\mathbf{l}$, are used to represent a vector.

B. Adversary Model

This paper assumes a very powerful adversary, characterized by both passive and active features.

The adversary assumed in this work is a spatially- and frequency-unlimited eavesdropper. Indeed, it can eavesdrop any RF communication that occurs in the network, independently from the specific IoT device originating the communication or the frequency used to exchange messages. Further, we assume that the adversary can compromise a set of IoT sensors, either by tampering the devices or by cloning some of them. Then, it can actively use these compromised devices to launch active attacks.

In particular, the aim of the adversary is manifold. First, the adversary would like to learn which specific sensor(s) satisfies the query. Second, the adversary could be interested in poisoning the protocol, by leading the network to compute a wrong value (e.g., in case of a MAX query, it would like to lead the network to compute a wrong maximum value). Moreover, the adversary would like to obtain the result of the query itself.

We also assume that, at a given time, to know the specific IoT sensor(s) originating the data, the adversary could compromise the storage of the sink IoT node, gaining access to the pieces of information in its possession. While this move inevitably leads to the disclosure of the queried value, we highlight that, in this attack, the adversary aims to obtain information about the specific IoT sensor(s) originating the queried value. Overall, an adversary can tamper the IoT devices in several different ways. For instance, the adversary

TABLE I
NOTATION USED THROUGHOUT THE PAPER.

Notation	Description
G	Generic sink IoT node.
N	Number of IoT sensors.
s_i	generic IoT sensor.
a_i	generic aggregator.
k_i	value shared between the sensor node s_i and the higher level aggregator.
J	Number of bits in the sensed data.
B_i	Value acquired by the node s_i.
$b_{i,j}$	Bit j of the data sensed by the generic IoT sensor s_i.
$m_{i,j}$	Generic message j delivered by the IoT sensor s_i.
K	Replication factor, i.e., number of random bits generated for each sensed bit $b_{i,j}$.
v	Random value extracted by the sink IoT node for each instance of the protocol.
$\mathbf{r_{i,j}}$	Bit-string delivered by the node i for the bit j.
H	generic hashing function.
T	Time between consecutive rounds of an instance of the protocol.
I	Number of aggregators directly-connected to the sink IoT node G.
$\mathbf{s_{i,j}}$	Aggregated message as elaborated by an intermediate aggregator.
$\mathcal{A}$	Generic Adversary.
Φ	Set of sensor nodes that collected the maximum value in the network.
ω	Set of nodes compromised by an active adversary.
p_s	Success probability of PPRQ.
p_f	Probability that PPRQ fails.

could physically compromise the leaf IoT devices, gaining full access to the node. In addition, multiple IoT sensors could collude and try to poison the computation of the queried value. Throughout the rest of this paper, we will neglect the specific logic used to gain control of the leaf IoT devices, and we will talk about *compromised leaf IoT devices*.

In Section V we will demonstrate that, under the assumptions described above, the adversary is required to be in control of all the devices in possession of the queried information to achieve its objectives.

IV. THE PPRQ SCHEME

The PPRQ scheme presented in this section allows the sink IoT node to enforce confidential and privacy-preserving MAX/MIN range queries in large IoT networks, while also being robust against passive (external) and active (internal) adversaries. In Section IV-A, we first introduce the underlying mechanism behind PPRQ through a baseline example. Then, the full scheme is formally described in Section IV-B.

A. Baseline Example

In this subsection, we introduce the main logic of PPRQ through a baseline example, depicted in Figure 2.

We assume that PPRQ is triggered by the sink IoT node G, through a specific broadcast *query* message delivered to all the nodes in the network. Let us assume that the query is a *MAX* query, i.e., the sink IoT node would like to obtain the maximum value sensed by the sensors. At the reception of this message, each sensor s_i collects the information from the surrounding environment. Let us assume that the number of bits representing the sensed value (B_i) is $J = |B_i| = 3$. Then, for each bit j sensed by the node i, namely $b_{i,j}$, starting from the Most Significant Bit (MSB), each node generates a local vector $\mathbf{r_{i,j}}$ made up of K elements. Specifically, if $b_{i,j} = 0$, then $\mathbf{r_{i,j}} = \mathbf{0}$. Otherwise, if $b_{i,j} = 1$, the node generates K random bits, that are assigned to the vector $\mathbf{r_{i,j}}$. Note that K is a *replication factor*, whose aim is to increase the reliability of the protocol computation. This vector is then delivered to the next-level aggregator. Recalling the example in Figure 2, the leftmost bit sensed by the sensors s_1, s_2, and s_3 is 0, and therefore, at the first round $j = 1$, they send $m_{1,n} = \mathbf{r_{i,1}} = \mathbf{0}$ $(i = 1, 2, 3)$. Conversely, the first bit of the value sensed by s_4 is 1, hence s_4 generates $K = 3$ random bits—let us assume $\{0, 1, 1\}$—, and it sends $m_{1,4} = \mathbf{r_{4,1}} = (011)$ to the aggregator a_2.

The aggregators perform the logical bitwise xor of the received values and forward them to the next level. Therefore, recalling the example in Figure 2, the node a_1 performs the operations $000 \oplus 000 = 000$, while the aggregator node a_2 executes $000 \oplus 011 = 011$, and both a_1 and a_2 send the resulting values to the sink IoT node G. Besides, if an aggregator obtains a vector $\mathbf{r_{i,j}} \neq 0$, it marks as *inactive* all the nodes that delivered a vector $\mathbf{r_{i,j}} = 0$. In this example, the aggregator node a_2 marks as inactive the link towards s_3, given that it computes a vector 011. Instead, a_1 does not mark any node as inactive, given that the vector it computed does not contain any 1. Marking a link as *inactive* means that, independently from the specific values it will receive on that link in the following rounds, it will always consider the received bit-string as 000.

Then, G performs a new xor operation, $000 \oplus 011 = 011$. Now, if the resulting bit-string is 000, the sink IoT node decides for the maximum to be 0. Otherwise, if there is at least a 1 in the resulting bit-string, it decides that the maximum is 1. In this case, given that the resulting bit-string is 011, it decides that the maximum value is 1. Besides, on the computation of a 1, the sink IoT node marks as *inactive* any link that delivered a bit-string with all 0. In this case, it marks as inactive the link connecting to the aggregator a_1. It means that independently from the specific values it will receive on that link in the following rounds, it will always consider the bit-string coming

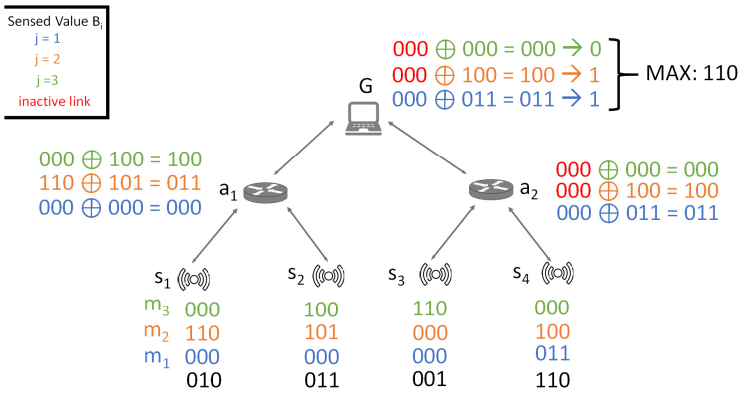

Fig. 2. Example of an instance of PPRQ with J messages ($J = 3$ and $K = 3$). Values in black color represent the sensed values, while the first, second, and third round of the protocol are represented by the blue, orange, and green color, respectively.

from that link as 000.

Therefore, in the second round of the protocol ($j = 2$), despite the value delivered by a_1 is 001, it will consider that value to be 000. Given that a_2 delivered 100, the resulting value is $000 \oplus 100 = 100$, and thus the sink IoT node decides that the maximum value is 1. Similarly, at the third round of the protocol ($j = 3$), the sink IoT node nullifies the string delivered by a_1 to 000, and it performs $000 \oplus 000 = 000$, and then it decides that the maximum of this bit is 0. Therefore, the maximum is 110, that is, indeed, the maximum value sensed by the four IoT sensors.

We emphasize that despite the version presented in the example above requires exactly $J = 3$ messages from the IoT sensors, the number of bits $K = 3$ required for each of the J bits sensed by the IoT devices can be easily aggregated in a single wireless message (provided that they do not exceed the related Maximum Transmission Unit (MTU)), further reducing the bandwidth cost. The process is shown in Figure 3.

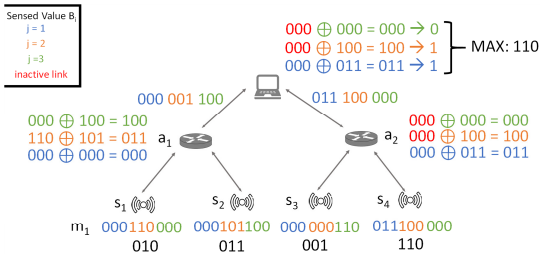

Fig. 3. Example of an instance of PPRQ, with a single wireless message (while $J = 3$ and $K = 3$). Values in black color represent the sensed values, while the values for the first, second, and third bit extracted by the IoT sensors are represented by the blue, orange, and green color, respectively.

In this case, each message delivered by the sensors and the aggregators has exactly $K \cdot J$ bits. In turn, each aggregator evaluates the bits of the messages in groups of K bits, and it nullifies all the following bits coming from a given IoT sensor, in case there is a group of K bits set to **0** that are xor-ed with a group of K bits containing at least a single 1.

We also notice that the sink IoT node only knows the maximum value sensed by the IoT sensors, but it does not know which specific sensor(s) produced that value and not even its location in the network. It only knows the next-level aggregators providing this value.

This baseline scheme presented in this section will be extended in the following Subsection to: (i) achieve authentication of the delivered messages; and, (ii) protect against passive eavesdropping.

B. Details of the PPRQ Scheme

PPRQ extends the mechanisms described in the previous subsection, by adding lightweight messages authentication and confidentiality services. The steps required by PPRQ are detailed below.

- The protocol is triggered by the sink IoT node, either upon request of a range query by the Internet or on a regular time basis. To initiate the protocol, the sink IoT node broadcasts a *query* message. The message contains: (i) the type of range query (max, min, or range), (ii) a random number v, and (iii) eventual optional information, such as the details of the range query. We assume that this message is authenticated by the sink IoT node, e.g., via a lightweight delayed-authentication technique such as TESLA and its variants, not requiring any symmetric cryptography technique [47]. The following discussion assumes the computation of *MAX* range queries, while *MIN* queries and range queries can be obtained as described previously in Section III-A. Note that the *aggregators* re-broadcast the message received from the sink IoT node down to the tree, up to the IoT sensors.
- On receiving the *query* message, the generic sensor node s_i first sets a timer, T_R. Every time this timer expires, the sensors run a round $j \in [1, J]$ of the protocol. We notice that the setup of the timer is optional: indeed, in case the protocol is setup to deliver a single message, the timer is not necessary. Then, the sensor s_i acquires the desired value from the surrounding environment, via its sensing capabilities. Let us name the acquired value as B_i, and assume that B_i is a binary bit-string, made up of J bits, namely $B_i = [b_{i,1}, \ldots, b_{i,j}, \ldots, b_{i,J}]$ (note that the number of rounds of the protocol is equal to the number of bits in the binary bit-string, and thus they are denoted with the same symbol J). Then, for each bit $b_{i,j}$ of the sensed

value, the sensor node s_i generates a vector with K values, namely $\mathbf{r_{i,j}} = [r_{i,j,0}, r_{i,j,1}, \ldots, r_{i,j,k}, \ldots, r_{i,j,K}]$. If the bit $b_{i,j}$ is a 0, then $\mathbf{r_{i,j}} = 0$; otherwise, the node i extracts K random bits, and inserts them in the vector $\mathbf{r_{i,j}}$. Then, the sensor node i executes the operation in Eq. 1.

$$\mathbf{s_{i,j}} = H(v||k_i) \oplus \mathbf{r_{i,j}}, \tag{1}$$

where v is the random number received from the sink IoT node, k_i is the random value in possession of the node i, the notation $H(\cdot)$ refers to a generic hashing function (having output size of K bits), and, finally, $\oplus$ refers to the bitwise X-OR operation. Then, the generic IoT sensor node s_i delivers this value to the next-hop aggregator IoT node in a dedicated *information message*. Note that in case the protocol is set up to deliver a single message, all the rounds j are executed consecutively, and the resulting $K \cdot J$ bits are delivered in a single message to the next-hop aggregator.

- On receiving an *information message* from a leaf sensor device, a generic *aggregator* a_i temporarily holds the message, until one of the following two conditions are met: (i) it receives the *information messages* from all the leaf devices connected to the aggregator, or (ii) a time-out T expires. Note that, in case the time-out expires, the aggregator assumes that a value $\mathbf{r_{i,j}} = 0$ has been received from the node i. Then, the aggregator node a_i first obtains the vector $\mathbf{r_{i,j}}$ delivered by the sensor i. To this aim, for each received value, it executes the operations in Eq. 2.

$$\mathbf{r_{i,j}} = \mathbf{s_{i,j}} \oplus H(v||k_i). \tag{2}$$

We recall that k_i is a random value shared between each couple of directly-connected IoT sensor and aggregator. Moreover, the value v is the random value extracted by the sink IoT node for each instance of the protocol, and delivered in broadcast to any node in the network through the first message of the protocol, namely the *Query Message*. Then, considering all the values received from the leaf sensors and the local value k_i, the aggregator node a_i generates a new aggregated message $\mathbf{s_{i,j}}$, according to the operation in Eq. 3.

$$\mathbf{s_{i,j}} = H(v||k_i) \oplus \left(\oplus_{i=1}^{I} \mathbf{r_{i,j}}\right), \tag{3}$$

where I is the number of leaf sensors directly connected to the aggregator a_i. Then, the aggregator a_i delivers this value to the next-hop aggregator IoT node in a dedicated *information message*.

- If the value of the bit-string $\left(\oplus_{i=1}^{I} \mathbf{r_{i,j}}\right)$ computed by the aggregator contains at least a 1, the aggregator marks as *inactive* all the links that delivered an aggregated value $\mathbf{r_{i,j}} = 0$. Note that this operation can be performed by the aggregator, since it owns the random values of the nodes directly-connected, and thus it can isolate their single contributions. Therefore, for each directly-connected link q, with $q = (1, \cdots, Q)$, the aggregator executes the operation in Eq. 4.

$$\mathbf{r_{q,j}} = \mathbf{s_{q,j}} \oplus H(v||k_q), \tag{4}$$

Marking a link as *inactive* means that, in all the following rounds of the protocol, the aggregator will assume that the value $\mathbf{r_{q,j}}$ delivered received on that link is $\mathbf{r_{q,j}} = 0$, independently from the specific value received on that link. We remark that in case the protocol is configured to take a single round, marking a link as *inactive* means that all the remaining $K \cdot j$ bits in the bit-string delivered by the specific IoT sensor are set to $\mathbf{0}$.

- The operations described in the previous step are repeated for each aggregator in the topology, up to the sink IoT node. The sink IoT node can obtain the final aggregated vector $\mathbf{r_{i,j}}$ by applying locally Eq. 5, leveraging the random values k_i shared with the directly-connected aggregators.

$$\mathbf{r_{i,j}} = \mathbf{s_{i,j}} \oplus_{i=1}^{I} H(v||k_i). \tag{5}$$

Note that each bit in the final vector $\mathbf{r_{i,j}}$ is the result of multiple bitwise xor operations. We recall that, according to the logical xor operation, the result on the single bit is a 1 if the input is an *even* number of 1s, while the result is a 0 if the number of 1s in the input is 0 or an *odd* number. Therefore, based on the nature of the bitwise xor and the randomization logic explained before, if at least one bit in the vector $\mathbf{r_{i,j}}$ is 1, the maximum value of the bit j is undoubted $m_j = 1$. Conversely, if all the bits in the vector $\mathbf{r_{i,j}}$ are set to 0, the sink IoT node will assume that the maximum value of the bit j is $m_j = 0$. We notice that in case the maximum value is a 0, given that $0 \oplus 0 = 0$ and the node does not apply the randomization, the sink IoT node cannot have a 1. Therefore, the protocol cannot lead to a false-positive event, i.e., a 0 that is interpreted as a 1 on the sink IoT node. Conversely, in case the maximum value is 1, given that $1 \oplus 1 = 0$, it is possible that the protocol leads the sink IoT node to compute a false-negative, i.e., a 1 interpreted as a 0. The likelihood of this event can be reduced thanks to the randomization process applied by the sensor nodes. Specifically, the higher the value K, the higher the chances that at least one bit in the vector $\mathbf{r_{i,j}}$ is 1, leading to the correct computation of the *max* value. In Section VI we will provide the details of the performance of the protocol while varying the value of K.

- If the value of the bit m_j computed by the sink IoT node is a 1, the sink IoT node marks as *inactive* all the links that delivered an aggregated value $\mathbf{r_{i,j}} = 0$, in the same way the aggregators do for the IoT sensors directly connected.
- If the protocol is configured to deliver a single message for each round j, at the expiration of the timer T_R previously set, the sensors start a new round of the protocol, by sending the next bit of the bit-string representing the collected value. For each round j of the protocol, the sensors behave in the same way, as described before. The only difference is in the behavior of the aggregators and the sink IoT node, based on the selection of inactive links described at the previous steps. We stress that this operation is not strictly necessary, as the protocol can

likely conclude in a single round, by inserting all the $K \cdot J$ bits in a single message.

For the readers' convenience, we report the pseudo-code of the operations executed by the IoT sensors, the aggregators, and the sink IoT node in Algorithm 1, Algorithm 2, and Algorithm 3, respectively, considering the case where a single wireless message is delivered for each IoT sensor.

Input: Receive v and range query details from the sink IoT node.
Store v;
Sense $B_i = \left[b_{i,1}, \ldots, b_{i,j}, \ldots, b_{i,J}\right]$;
for $j \leftarrow 1$ **to** J **by** 1 **do**
  if $b_{i,j} == 0$ **then**
    r_i = $[0]^K$
  end
  else
    Extract $r_i \xleftarrow{\$} \{0,1\}^K$;
  end
end
$\mathbf{s_i} \longleftarrow$ H($v \parallel k_i$) $\oplus$ $\mathbf{r_i}$;
Output: Transmit $\mathbf{s_i}$.

Algorithm 1: Pseudo-code of PPRQ on the generic IoT sensor.

Input: Receive messages from the Q IoT sensors directly connected.
for $q \leftarrow 1$ **to** Q **by** 1 **do**
  $\mathbf{r_q} \longleftarrow \mathbf{s_q} \oplus$ H($v \parallel k_q$).
end
/* Aggregate messages. */
for $j \leftarrow 1$ **to** $K \cdot J$ **by** K **do**
  $\mathbf{s_i} \longleftarrow$ H($v \parallel k_i$) $\oplus$ $\left(\oplus_{q=1}^{Q} \mathbf{r_{q,j}}\right)$;
  if $\mathbf{r_{q,j}} == [0]^K \wedge \mathbf{r_{\tilde{q},j}} \neq [0]^K$ **then**
    /* Mark q as inactive. */
    $\mathbf{r_q} = [0]^K$
  end
end
Output: Transmit $\mathbf{s_i}$.

Algorithm 2: Pseudo-code of PPRQ on the generic aggregator a_i.

Input: Receive Q messages from the aggregators directly connected.
for $q \leftarrow 1$ **to** Q **by** 1 **do**
  $\mathbf{r_q} \longleftarrow \mathbf{s_q} \oplus$ H($v \parallel k_q$).
end
/* Extract the queried value */
for $j \leftarrow 1$ **to** $K \cdot J$ **by** K **do**
  $\mathbf{r}$ = $\oplus_{q=1}^{Q}$ $\mathbf{r_q}$;
  if $\mathbf{r_{q,j}} == [0]^K \wedge \mathbf{r_{\tilde{q},j}} \neq [0]^K$ **then**
    /* Mark q as inactive. */
    $\mathbf{r_q} = [0]^K$
  end
end
Output: $\mathbf{r}$ is the queried value.

Algorithm 3: Pseudo-code of PPRQ on the sink IoT node.

We notice that the sink IoT node is the only node that knows the *maximum* (or, in general, the queried) value sensed by the IoT sensors. However, it does not know which specific IoT sensors sensed that value, not even the number of sensors that sensed that value nor the location of that sensors. The only information it has is the next-level aggregator that provided that value. Therefore, as will be discussed in the following section, the proposed PPRQ protocol emerges as a lightweight and privacy-preserving solution to IoT networks querying.

V. Security Analysis

In the following, we formally describe the security features offered by the PPRQ protocol. Specifically, we distinguish between an *external* passive eavesdropper, discussed in Section V-A, and an *internal* active attacker, discussed in Section V-B.
To ease discussion, we assume that the protocol is set up with a single message for each round j. However, the security considerations discussed below are applicable also when the protocol requires a single message for each IoT sensor.

A. Passive Adversary

Let us assume that the adversary is a passive eavesdropper, as outlined in Section III-B, interested in: (i) obtaining the local readings of the sensor; and, (ii) computing the maximum reading value. In Proposition V.1, we show that PPRQ is robust against such an attacker, and protects the *query privacy*.

Proposition V.1. *Let us assume that $\mathcal{A}$ is an external passive eavesdropper. Then, under standard security assumptions, $\mathcal{A}$ is prevented from formulating an educated guess on the value of the local readings of the sensor $r_{i,j}$, or the maximum reading value.*

Proof. Let us assume that $\mathcal{A}$ passively eavesdrops the traffic through an IoT node s_j. Each node s_i ($j \neq i$), being it an aggregator or an IoT sensor, transmits on the wireless channel a value consistent with the following Equation 6.

$$\mathbf{s_{i,j}} = H(v||k_i) \oplus \mathbf{r_{i,j}}, \tag{6}$$

where v is a random value broadcasted by the sink IoT node, k_i is the bit-string value shared between the transmitting node and the sink IoT node, and $\mathbf{r_{i,j}}$ is the *information*, possibly being the bitwise xor of other information coming from the lowest level of the tree.
We notice that $\mathcal{A}$ could read the value $\mathbf{r_{i,j}}$ using two strategies. First, the adversary can read the value $\mathbf{r_{i,j}}$ if it possesses the value k_i. However, the standard security assumption is that each node maintains its local value of k_i private.
Alternatively, the adversary can read the value $\mathbf{r_{i,j}}$ if it can reverse the result of the hashing function $H(\cdot)$. Under standard security assumptions, a hashing function cannot be inverted [48].
Further, the above considerations apply also for the MAX/MIN sensor readings, i.e., the queried values. Therefore, PPRQ is robust and privacy-preserving against any passive eavesdropper, given that the adversary cannot perform an educated guess on the maximum value collected by the sensors. □

B. Active Adversary

Let us now consider an active adversary ($\mathcal{A}$) that has compromised a set of sensor nodes (ω). Its aim could be manifold: (i) to obtain the maximum value sensed in the network; or, (ii) to pollute the protocol, by leading the sink IoT node to compute a value different from the real maximum sensed by the IoT devices.

Let us define with Φ the set of sensor nodes that collected the maximum value in the network.

Proposition V.2. *If $\Phi = \omega$, the adversary $\mathcal{A}$ is always capable of disrupting the reading of the MAX/MIN value.*

Proof. The proof of this proposition is straightforward. Indeed, if the set of compromised IoT sensors ω includes all the nodes in Φ, $\mathcal{A}$ can modify the readings of the sensors and prevent the MAX/MIN from being communicated. □

Despite the result of the above proposition, we should note that this event is quite unlikely in regular operation conditions, given that the set of sensors sensing the maximum value usually changes dynamically and (with some degree) also unpredictably. Therefore, to maximize the probability of being in this favorable situation, the adversary would be forced to compromise as many IoT sensors as possible.
Let us now assume that $\omega \subset \Phi$, i.e., the MAX/MIN value has been sensed by a set of sensors, where not all of them have been compromised by the adversary. In this more realistic assumption, we investigate the chances for the adversary to be able to modify the result of the query.
Note that, to modify the result of the query, the adversary has to guess the value of the bit-string $\mathbf{r_{i,j}}$ originating from the IoT sensors that are not in ω. Therefore, the following Proposition V.3 applies.

Proposition V.3. *An adversary $\mathcal{A}$ that compromises a proper subset of the IoT sensors $\omega \subset \Phi$ cannot control the maximum value sensed in the network.*

Proof. The proof of this proposition follows the proof of the Proposition V.1. Specifically, let us assume the most favorable case for the adversary, where $\mathcal{A}$ controls all the IoT sensors in Φ but 1, that is the IoT sensor s_i. In this case, from the non-compromised IoT sensor, $\mathcal{A}$ could eavesdrop a value in line with Eq. 6, that is $H(v||k_i) \oplus \mathbf{r_{i,j}}$.
Given that $\mathcal{A}$ does not know the value k_i, any of the 2^{k_i} possible values has the same probability to appear, that is $p = \frac{1}{2^{k_i}}$. Thus, either transmitting 0 or 1 leads the adversary to the same success probability, that is $p = \frac{1}{2}$. Therefore, the adversary cannot guess whether to send through its compromised node(s) either $\mathbf{r_{i,j}} = 0$ or any $\mathbf{r_{i,j}} = 1$. This proves that $\mathcal{A}$ could not learn nor control the maximum value sensed in the network, and thus, that PPRQ protects the *query privacy*. □

Pollution attack. As per the pollution of the results computed by the protocol, we focus on two attacks that the adversary could try to perform: (i) injecting a false-positive; or, (ii) injecting a false-negative.
We first consider the case of a false-positive, where $\mathcal{A}$ injects a 1 instead of sending the correct value 0. In this case, the following Proposition V.4 applies.

Proposition V.4. *Given $\Phi \neq \emptyset$, an active adversary $\mathcal{A}$ can always pollute the protocol by injecting a MAX (*false-positive*).*

Proof. The proof of this proposition is straightforward. Let us assume that the adversary $\mathcal{A}$ compromises at least a single sensor s_i, and that sensor delivers a value that is the *MAX* of the overall sensing range. If a sensor s_i provides a value that is the *MAX* of the overall sensing range, it will always result as the MAX value of the overall network. □

In this scenario, the network administrator can identify the attack if he/she identifies that the sensors in ω have been compromised. This can be done through some audit activities, that can be triggered, e.g., when the network administrator realizes that there is a node delivering outliers values. For comparison, we remark that none of the protocols currently available in the literature can inherently detect and prevent this attack. In addition, we note that the above strategy is ineffective when the adversary is interested in acquiring the real *MAX* value sensed in the network.
Let us now consider the case of the false-negative, i.e., the adversary would like to inject a 0 when, instead, the real sensed value is a 1. In this case, the following Proposition V.5 applies.

Proposition V.5. *Let us assume that $\Phi \subset \omega$. In this case, the adversary $\mathcal{A}$ has a success probability of injecting a false-negative that is bounded by:*

$$p_s = 1 - \left[1 - \left[1 - \left(\frac{1}{2}\right)^K\right]^J\right]^T$$

.

Proof. In line with the hypothesis in Proposition V.5, let us assume that there is a single IoT sensor s_i s.t. $s_i \in \omega$ and $s_i \notin \Phi$. This assumption means that the adversary controls all the nodes in ω but s_i.
To inject a false-negative, $\mathcal{A}$ should be able to insert a 0 if $r_{i,j} = 0$, or it should insert a 1 if $r_{i,j} = 1$. However, due to Proposition V.1 and Proposition V.3 and related proofs, $\mathcal{A}$ does not know $r_{i,j}$.
Without loss of generality, let us also assume that T nodes out of N sense the maximum value, and the adversary compromises all but one of these nodes. Considering a single IoT sensor, the probability that PPRQ fails is expressed by the following Eq. 7.

$$p_f = 1 - [p_{s,N=1}]^J = 1 - [p_{s,N=1}]^J = 1 - \left[1 - \left(\frac{1}{2}\right)^K\right]^J \tag{7}$$

Overall, PPRQ ends successfully if at least one of the T nodes in possession of the MAX delivers its value successfully. Therefore, PPRQ is successful if none of the T nodes that have the MAX fails, according to the following Eq. 8.

$$p_s = 1 - \left[1 - \left[1 - \left(\frac{1}{2}\right)^K\right]^J\right]^T \tag{8}$$

This proves Proposition V.5, and it also proves that even controlling all nodes but one, the active adversary $\mathcal{A}$ cannot neither learn the queried value nor impose a specific value. □

Considerations on Aggregators Tampering. The discussion above assumes that the attacker can physically compromise one or more IoT sensors, and not the aggregators. This is a

common and reasonable assumption, as usually IoT sensors are low-cost devices, more vulnerable and, in general, easier to compromise.

To provide further insights, we hereby include some more considerations on the possibility that the adversary compromises one or more aggregators. In the following, to ease discussion, we assume that there is only a single IoT sensor reporting the queried value, and that the probability for the queried value to occur on any IoT sensor is uniformly distributed.

Proposition V.6. *Let us assume there are $N = 2^L$ IoT sensors in the network, arranged according to a binary tree topology with L layers. An adversary $\mathcal{A}$ that compromises a subset of aggregators ω_l on a given layer l, with $l \leq L$ and $\omega_l \leq 2^{L-l}$, can guess the queried value with a probability $p_g = \frac{2^l \cdot \omega_l}{N} = 2^{l-L} \cdot \omega_l$.*

Proof. To prove our proposition, let us assume the example network topology reported in Figure 4.

Assume $\mathcal{A}$ compromises an aggregator at the layer $l = 1$

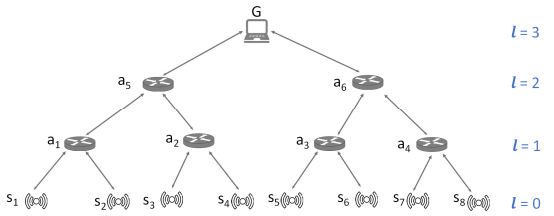

Fig. 4. Reference Scenario with $N = 8$ for the discussion on the tampering of aggregators.

of the topology, e.g., a_1. Given that a_1 is in the possession of the values k_1 and k_2 shared with s_1 and s_2, respectively, $\mathcal{A}$ can access the value of the respective readings. Similar considerations apply for any other aggregator on the layer $l = 1$ of the topology. Therefore, assuming $\mathcal{A}$ compromises ω_1 aggregators on the layer $l = 1$, its guessing probability will be $p_g = \frac{2*\omega_1}{N}$, i.e., $p_g = \frac{2*\omega_1}{2^L}$. Note that $\mathcal{A}$ can know with 100% accuracy ($p_g = 1$) the queried value only if it controls all the aggregators on the layer $l = 1$, i.e., $\omega_1 = 2^l$. Generalizing the discussion for the generic l-th layer, we have that $p_g = \frac{2^l \cdot \omega_l}{N} = 2^{l-L} \cdot \omega_l$.

□

Regarding *location privacy* when a subset of aggregators are compromised, there are several cases, based on the layer where the compromised aggregators are located. In the following, we discuss two main cases, referring to the topology in Figure 4.

Case 1: Aggregators directly-connected to IoT sensors. In case $\mathcal{A}$ compromises an aggregator that is directly connected to the IoT sensors, given that the aggregator has the value k_i shared with the directly-connected IoT sensors s_i, $\mathcal{A}$ has direct access to the readings of the directly-connected sensors s_i. However, note that $\mathcal{A}$ does not have access to the readings of the other IoT sensors. For instance, let us assume that $\mathcal{A}$ compromises all the aggregators $[a_1, a_2, a_3]$, but it does not compromise a_4. In this case, $\mathcal{A}$ has access to the values $\mathbf{r_{i,j}}$ delivered by the IoT sensors $s_1, s_2, s_3, s_4, s_5, s_6$ in Figure 4. Further, based on the previous Proposition V.1 and Proposition V.6, $\mathcal{A}$ cannot know $\mathbf{r_{7,j}}$ and $\mathbf{r_{8,j}}$, i.e., the plain-text information delivered by the IoT sensors s_7 and s_8. However, should $\mathcal{A}$ control also the additional aggregator a_4, it can know the location of the IoT sensor generating the queried value. Therefore, for this case, similar considerations to the ones in the Proof of Proposition V.6 apply, as described in the following Proposition V.7.

Proposition V.7. *Let us assume there are $N = 2^L$ IoT sensors in the network, arranged according to a binary tree topology with L layers. An adversary $\mathcal{A}$ that compromises a subset of aggregators ω_1 on the layer $l = 1$ of the topology, with $\omega_l \leq 2^{L-l}$, can guess the location of the IoT sensor(s) generating the queried value with a probability $p_g = \frac{2 \cdot \omega_1}{N}$.*

Proof. Let us assume that $\mathcal{A}$ compromises a single aggregator a_i on the layer $l = 1$ of the topology. Given that a_i is in the possession of the values k_i shared with the directly-connected sensors, $\mathcal{A}$ can access the value of the respective readings. Similar considerations apply for any other aggregator on the layer $l = 1$ of the topology. Therefore, assuming $\mathcal{A}$ compromises ω_1 aggregators on the layer $l = 1$, its guessing probability on the location of the IoT sensor(s) generating the queried value will be $p_g = \frac{2*\omega_1}{N}$, i.e., $p_g = \frac{2*\omega_1}{2^L}$. Note that $\mathcal{A}$ can know with 100% accuracy ($p_g = 1$) the location of the IoT sensor(s) generating the queried value only if it controls all the aggregators on the layer $l = 1$, i.e., $\omega_1 = 2^{L-1}$. □

Case 2: Aggregators not directly-connected to IoT sensors. In case $\mathcal{A}$ compromises one or more aggregators that are not directly connected to the IoT sensors, it can acquire only limited information about the IoT sensor(s) generating the queried value. With reference to Figure 4, for instance, assuming $\mathcal{A}$ compromises a_5, it only learns the aggregated values delivered by the lower-layer aggregators a_1 and a_2. However, based on the previous Proposition V.1 and Proposition V.6, $\mathcal{A}$ cannot know which values are reported by the aggregators connected to the IoT sensors in the other parts of the network (s_5 to s_8). In addition, note that the values delivered by a_1 and a_2 are aggregated from the lower layers of the topology. Therefore, the following Proposition V.8 applies.

Proposition V.8. *Let us assume there are $N = 2^L$ IoT sensors in the network, arranged according to a binary tree topology with L layers. An adversary $\mathcal{A}$ that compromises all the aggregators ω_l on the layer l of the topology, with with $l \leq L$ and $\omega_l \leq 2^{L-l}$, can guess the location of the IoT sensor(s) generating the queried value with a probability $p_g = \frac{1}{2^{l-1}}$.*

Proof. Let us assume that $\mathcal{A}$ compromises a single aggregator a_i on the layer l of the binary tree topology. Given that a_i is in the possession of the values k_i shared with the directly-connected sensors, $\mathcal{A}$ can access the value of the readings originated from the layer $l + 1$ of the topology. Thus, $\mathcal{A}$ can know the specific aggregator on the layer $l - 1$ originating the queried value. However, such readings are aggregate from the layer $l + 2$ of the topology, where there are 2^{l-1} nodes. Note that the same considerations apply if $\mathcal{A}$ compromises more aggregators. Therefore, assuming that there is only a single IoT sensor reporting the queried value, and that the

probability for the queried value to occur on any IoT sensor is uniformly distributed, even compromising all the aggregators on a given layer l, the maximum probability for $\mathcal{A}$ to guess the location of the IoT sensor generating the queried value is $p_g = \frac{1}{2^{l-1}}$. □

Note that our discussion can be further generalized, to take into account the presence of multiple queried (MAX/MIN) value, a not-uniform distribution (with respect to the IoT sensors) of such values, or the possibility that $\mathcal{A}$ compromises aggregators on different layers. This is left for future work.

Final Considerations. From the discussion above, it results that the advantage for the adversary increases if the adversary can also compromise aggregators. On the one hand, we highlight that the above-highlighted limitation is the price to pay for the reduced computational overhead required by PPRQ on the IoT sensors and the aggregators. Indeed, when the IoT sensors and the aggregators cannot support cryptography functions and edge/fog nodes cannot be deployed due to limitations in the budget or the scenario, to the best of our knowledge, PPRQ is the only solution that can still guarantee a non-zero level of *query accuracy*, *query privacy*, and *location privacy*, even assuming an adversary that can compromise all the IoT sensors, but one, and all the aggregators on a given layer of the topology, but one.
On the other hand, being aware of these limitations, the system administrator can deploy dedicated additional protection schemes to protect the aggregators from physical tampering. Specifically, to cope with this threat, we can resort to two solutions. The first one is to deploy tamper-proof aggregators (or tamper-evident ones, if it is enough to detect that a compromise happened), such as the ones described in [49] and [50]. The second (partial) solution would be to make the nodes indistinguishable by the adversary, so that it cannot make any educated guess about which nodes are the aggregators. Note that the adversary could recover the topology of the network via traffic analysis, hence it would be mandatory to implement a software layer that, at the expenses of further communications overhead, routes the messages in a way that weakens the relationship originator-router, such as the ones in [14], [51], [52], [53]. However, even in this latter case, it should be noted that the advantage of the adversary would increase with the number of nodes tampered with. Indeed, the more nodes are compromised, the higher the chance that an aggregator would be among them and, the more aggregators, the more the knowledge of the adversary over the sensed MAX/MIN and the corresponding sensing node.

VI. Performance Assessment

In this section, we report some results on the efficiency and robustness of the PPRQ scheme. Section VI-A provides the initial analytical model of PPRQ, applicable for low values of J. Section VI-B provides the results of some simulations of the protocol with higher realistic values of J, assuming both a benign and a malicious scenario. Finally, Section VI-C reports a qualitative and quantitative comparison of PPRQ to existing works on privacy-preserving range queries in IoT networks. For all the simulated results, we have implemented PPRQ within the software MATLAB R2020a, using a Dell XPS15 9560 Laptop, equipped with an Intel Core i7700HQ processor working at 2.80GHz, 32 GB of RAM, and 1TB of hard disk. Then, we used this implementation to evaluate the performance of PPRQ, both in a benign and malicious scenario.

A. Probabilistic Model of PPRQ

In the following, we derive an analytic model of the performance of PPRQ, based on sound concepts of probability theory. We anticipate that the following model provides a lower bound on the success probability of PPRQ, allowing to model the worst-case scenario for the performance of the protocol. However, the following discussion is very useful to fully catch the relationship among the different system parameters, such as the number of IoT sensors N, the replication factor K, and the number of sensed bits J on the overall success probability of PPRQ.

Let us consider the toy case where a single IoT sensor ($N = 1$), sensing a single bit ($J = 1$). In the case this value is 0, the protocol is always successful. In case the sensed bit is a 1, the IoT sensor extracts K bits, and PPRQ is successful if at least one of the extracted bits is a 1. Therefore, the lower bound of the success probability of PPRQ, in this case, can be modeled according to the following Eq. 9.

$$p_s \geq 1 - \left(\frac{1}{2}\right)^K. \tag{9}$$

Now, considering also $J > 1$, we notice that each of the j bits should be delivered successfully. Given that the processing of each bit is independent from the other ones, the following Eq. 10 holds.

$$p_s \geq \left[1 - \left(\frac{1}{2}\right)^K\right]^J. \tag{10}$$

Let us now take into account a generic IoT network with $N > 1$ nodes, organized in a binary tree topology (in line with the example in Section IV-A). The worst case, introduced in the following, is a bit counter-intuitive and it occurs when all the aggregators on a sensor-to-sink path xor strings that are different from zeros. Indeed, assuming an IoT sensor senses a bit with value 1, this value is routed to the sink IoT node if at least one of the K randomly extracted bits is set to 1. The only event that could prevent this from happening is when either all the K generated bits are zero, or an aggregator xors the resulting string (t) of K bits (having at least one element different from 0) with another one (t'), resulting in a vector of zeros (i.e., $t = t'$). This latter case only happens when the $t' \neq 0$, i.e., some of the leaves nodes underlying the branch t' have sensed a value that is different from 0. Applying recursively the exposed reasoning brings us to the worst case, i.e., when all the aggregators on a sensor-to-sink path xor strings that are different from zeros. That is, all the leaves are sensing a MAX, and that MAX is composed of J bits set to 1.

Thus, the information has to travel correctly exactly $log_2(N) + 1$ layers of the tree (including the first *local* step, where the data is sensed from the environment) to arrive

correctly at the destination. Therefore, the lower bound of the success probability of PPRQ can be expressed according to the following Eq. 11.

$$p_{s_{N,K,J}} \geq \left[1 - \left(\frac{1}{2}\right)^{K}\right]^{(log_2 N+1)\cdot J} \quad ; \qquad (11)$$

Note that the discussed scenario is the worst-case: indeed, if less than N IoT sensors sense the queried value, they will report always a vector $(0)^K$, and they will not impact negatively on the success probability.

An example for the reference values $J = [8, 16]$ and $N = [8, 16]$ is reported in Figure 5.

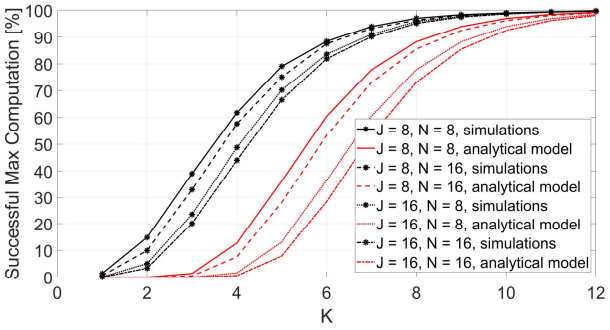

Fig. 5. Success Probability of PPRQ with generic values of K, assuming the case of $J = [8, 16]$ and $N = [8, 16]$, using simulations and the analytical model derived from Eq. 11

.

We notice that the most significant parameters of the model are K and J. Indeed, the success probability of PPRQ increases significantly while increasing K, while it decreases while increasing J. Therefore, when there are a significant number of bits sensed by the IoT sensors, increasing the replication factor K mitigates the potential additional error probability introduced by J. At the same time, we notice that the number of IoT sensors N impacts more slightly on the success probability (according to a logarithmic law).

Overall, we can notice that the number of nodes sensing the queried value has a non-negligible impact on the model, and it significantly affects the precise success probability of PPRQ. However, the derivation of the exact model of the success probability is out of the scope of this paper, and it is left as future work. The performances of PPRQ with reference values of J, N, and K are reported via simulations in the following Subsection.

B. Simulations

In the following, we study the performance of PPRQ concerning its system parameters, both in a benign and in an adversarial scenario.

Benign Scenario. Considering a benign scenario (no adversary), we evaluated the success probability of PPRQ, i.e., the percentage of experiments where the protocol converged to the correct value of the *MAX*. We tested the protocol in different operating conditions, i.e., while varying the number of IoT sensors in the scenario, the number of bits J used to represent the sensed value, and the value of K.

The following figures 6, 7, 8, and 9 show the overall success percentage of the protocol, considering $J \in \{8, 16, 24, 32\}$, $K \in \{2, 4, 6, 8, 10, 12\}$, and a number of IoT sensors $N \in \{8, 32, 128, 512\}$. Finally, note that each point in the above-cited figures has been obtained as the mean of $10,000$ tests, assuring that the reported values include the 95% confidence interval.

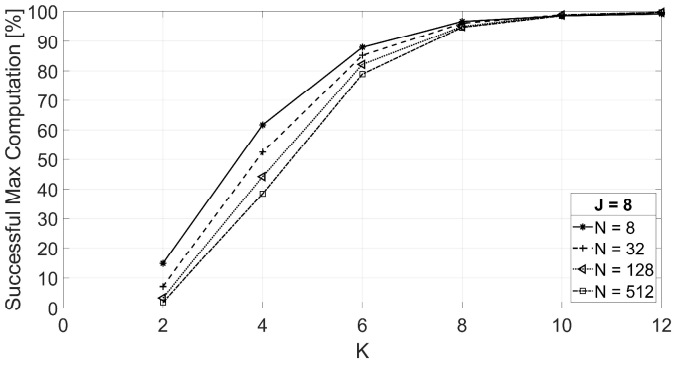

Fig. 6. Success Probability of PPRQ in a benign scenario, when the sensed data has $J = 8$ bits, by varying the number of replicas K of each bit and the number of nodes N in the scenario.

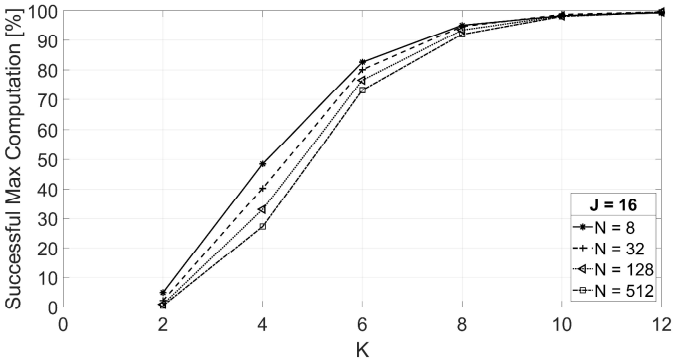

Fig. 7. Success Probability of PPRQ in a benign scenario, when the sensed data has $J = 16$ bits, by varying the number of replicas K of each bit and the number of nodes N in the scenario.

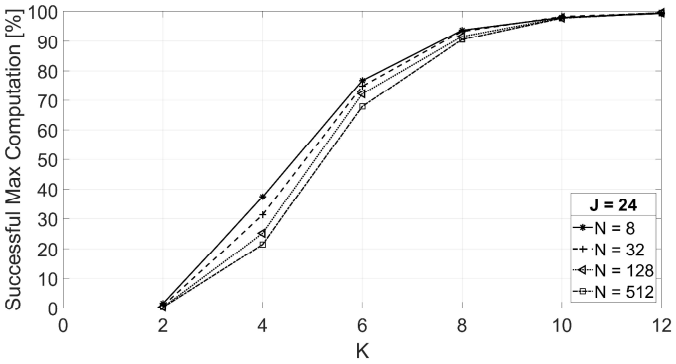

Fig. 8. Success Probability of PPRQ in a benign scenario, when the sensed data has $J = 24$ bits, by varying the number of replicas K of each bit and the number of nodes N in the scenario.

In line with our discussion in the previous subsection, we highlight the positive effect of increasing the value of K on the

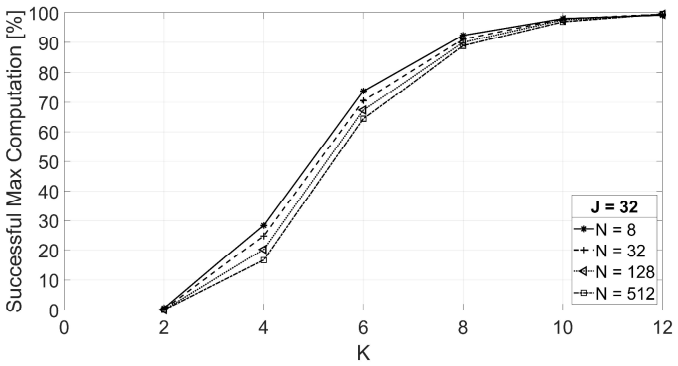

Fig. 9. Success Probability of PPRQ in a benign scenario, when the sensed data has $J = 32$ bits, by varying the number of replicas K of each bit and the number of nodes N in the scenario.

overall success probability of PPRQ. Indeed, independently from the number of IoT sensors in the scenario and from the number of bits to be delivered, increasing K significantly improves the success probability, i.e., the probability that the sink IoT node computes a maximum value that equals the *MAX* between all the readings of the IoT sensors. We notice that the choice $K = 12$ bits always guarantees a success probability greater than 99.9%, contributing to making PPRQ very reliable. We also note that increasing K causes only a little increase in the bandwidth overhead, given that all the K bits can be transmitted together in a single packet. Considering that state-of-the-art communication technologies at the MAC layer, such as IEEE 802.15.4, have a payload size of 92 bytes [54], PPRQ can be configured in a very reliable fashion, by requiring either a single message to be transmitted for each IoT sensors, for each bit, or only a single overall message for each IoT sensor, in case $K \cdot J < 92$ bytes.

We also observe that, for small values of K ($K < 8$), networks with more IoT sensors are less accurate than small networks, including fewer IoT sensors. This result is evident only when $J > 2$, and thus it does not emerge from the analytical model described before. This result occurs because, for small values of K, the probability to extract K times a 0 when a 1 is sensed is not negligible. Moreover, considering the same value of K, the chances that this event occurs are slightly higher when more IoT sensors are involved, thus leading to an increased error probability. We also notice that such performance difference tends to decrease by increasing the value of K, becoming almost negligible starting from $K = 8$ bits.

Comparing all the figures 6-9, we also notice that the success probability for a given network size and value of K decreases with an increase of J. Indeed, as discussed in the previous subsection, the probability for the protocol to succeed is given by the probability that, for each of the J bits, the protocol can successfully compute the correct value. Therefore, the higher the number of bits, the higher the chances that at least a single bit is delivered flipped, thus leading to an incorrect *MAX* computation.

Overall, Figure 6, Figure 7, Figure 8 and Figure 9 provide evidence that selecting the replication factor $K \geq 12$ bits ensures that PPRQ ends successfully in over 99.9% of the cases, guaranteeing outstanding reliability. In turn, these features make PPRQ also a scalable solution, as increasing the number of leaf IoT devices N does not impact on the overall success probability and overhead of the protocol. Therefore, any implementation of the scheme should configure $K \geq 12$ bits. We remark that this configuration does not affect the message overhead and the energy consumption of PPRQ, as all the information can be always transmitted in a single IEEE 802.15.4 packet, as shown in the example in Figure 3 (see Section VI-C for more details).

Scenario with Active Adversary. In this paragraph, we consider a malicious scenario, where an active attacker compromises a subset of the nodes in the network, forcing them to transmit 0 instead of the true sensed value. We first consider the most generic scenario, where each node can sense a value uniformly at random in the interval $\left[0, 2^{J-1}\right]$. The following figs. 10 and 11 show the success probability of PPRQ in a network with $N = 8$ and $N = 32$ IoT sensors, respectively. For the simulations, we assumed the protocol is configured with $K = 12$, and we investigated the gain of the adversary when compromising an increasing percentage of the IoT sensors.

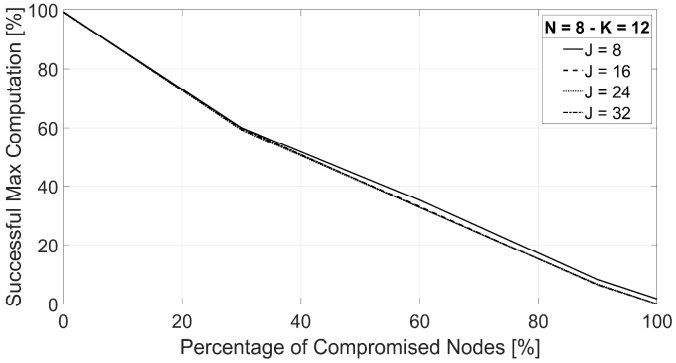

Fig. 10. Success Probability of PPRQ in a malicious scenario, with $N = 8$ IoT sensors and $K = 12$, when the adversary compromises an increasing percentage of the IoT sensors. Different line-styles report results with different values of J.

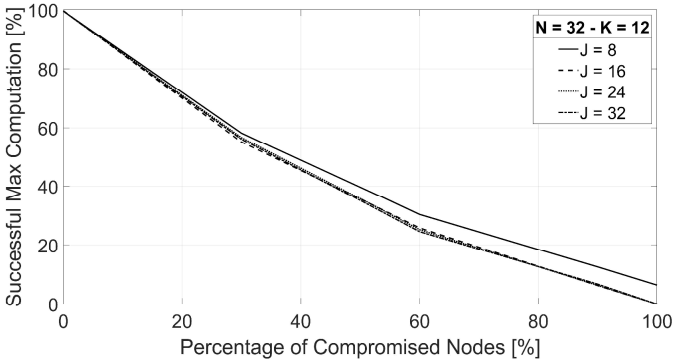

Fig. 11. Success Probability of PPRQ in a malicious scenario, with $N = 32$ IoT sensors and $K = 12$, when the adversary compromises an increasing percentage of the IoT sensors. Different line-styles report results with different values of J.

Overall, we notice that compromising a higher number of nodes increases the probability for the adversary to mislead

the protocol. This occurs because the more IoT sensors the adversary compromises, the higher the chances that the subset of compromised nodes (ω) includes the node(s) sensing the *MAX*, and therefore, the higher the chances for the adversary to succeed. In Figure 11, we also notice the higher robustness offered by the configuration with $J = 8$ compared to any configuration with higher values of J. As already described before, the overall success probability depends on the successful delivery of each of the J bits, and therefore, higher values of J provide the adversary higher chances to corrupt the computation. The cited phenomenon is hardly visible in Figure 10, since higher values for K (in this case, $K = 12$) do increase the success probability for the protocol, de-facto neutralizing in the analyzed graph the negative effect due to the increase in the number used to represent the sensed data (J).

Finally, in line with the security analysis reported in Section V, we investigated the overall performance of PPRQ when the adversary compromises an increasing percentage of the nodes sensing the *MAX*. This attack condition finds application when the adversary knows exactly which sensors are more likely to sense the *MAX* value. The results are reported in figs. 12, 13, and 14, assuming to have $N = 8$ nodes, and for $K = 4$, $K = 8$, and $K = 12$, respectively. As for the adversary, we assumed that the adversary compromises 0, 1, 3, 5, while 6 nodes out of the 8 sensing the MAX value.

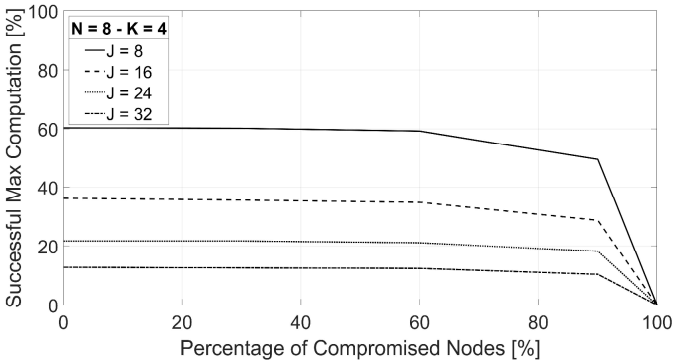

Fig. 12. Success Probability of PPRQ in a malicious scenario, with $N = 8$ IoT sensors and $K = 4$, when the adversary compromises an increasing percentage of the IoT sensors sensing the *MAX*. Different line-styles report results with different values of J.

We notice a different trend of the success probability for $K = 4$, compared to the cases of $K = 8$ and $K = 12$. When $K = 4$, the adversary has a slight advantage in compromising an increasing number of the nodes sensing the MAX. Conversely, when $K = 8$ and $K = 12$, we notice that compromising only one node or $N - 1$ nodes leads the adversary approximately to the same result, which is not different from the case where no nodes are compromised.

This result can be explained by recalling the security analysis in Section V and Theorem V.5. Evaluating Eq. 8, we can notice that, as expected, the higher the number of nodes that have the maximum, the higher the chances that PPRQ is successful. However, also K and J play a role. Indeed, the higher the ratio between K (or J) and T, the higher the success probability. This is the reason for the decay of the

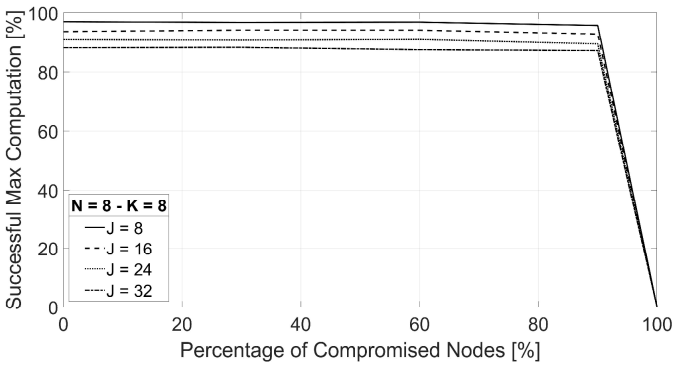

Fig. 13. Success Probability of PPRQ in a malicious scenario, with $N = 8$ IoT sensors and $K = 8$, when the adversary compromises an increasing percentage of the IoT sensors sensing the *MAX*. Different line-styles report results with different values of J.

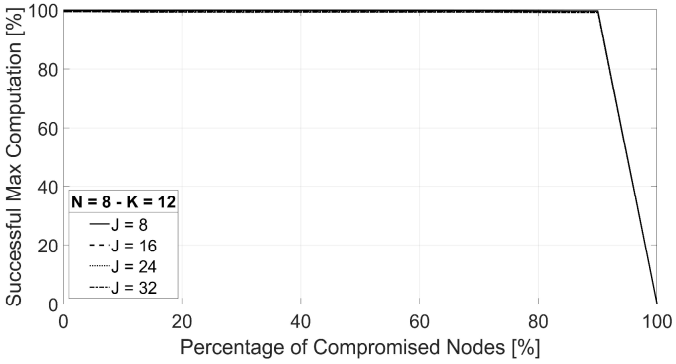

Fig. 14. Success Probability of PPRQ in a malicious scenario, with $N = 8$ IoT sensors and $K = 12$, when the adversary compromises an increasing percentage of the IoT sensors sensing the *MAX*. Different line-styles report results with different values of J.

success probability of PPRQ in Figure 12, where the values of K and J are little compared to T. Conversely, when $K > T$ or $J > T$, as in figs. 13 and 14, the adversary does not gain any evident advantage in compromising an increasing number of nodes, but it is successful only if it gains the possession of all the nodes in T.

Overall, only by compromising all the nodes sensing the *MAX*, the adversary can have the 100% assurance that the protocol fails, while the success probability of PPRQ remains at values close to 100% when up to $N - 1$ nodes sensing the *MAX* are compromised.

In summary, these results prove the extreme robustness characterizing PPRQ, even against powerful adversaries that can compromise a significant subset of the nodes in the scenario.

C. Comparison and Discussion

In this section, we compare PPRQ against competing solutions in the literature. First, we report in Table II a qualitative comparison between PPRQ and some of the works reported in Section II, along some reference system requirements. Note that we attached the tag *high* to scientific contributions that require either time-consuming and energy-demanding operations on the Aggregators, or the availability of significant storage

space. Conversely, we attached the tag *low* when the specific scientific contribution requires no encryption, and at maximum a single hashing operation from the aggregators.

We notice that, at the time of this writing, all the protocols that provide both *query privacy* and *location privacy* require high computational/storage effort on aggregator devices, usually implemented via edge/fog nodes. We highlight that an edge/fog node is usually a mains-supplied computing unit, that should be deployed in a location where fixed energy supply is available. Therefore, deploying edge/fog nodes is usually expensive, and it could be very challenging in remote scenarios, due to the unavailability of mains-supplied locations. In addition, all the protocols that require low computational/storage effort on aggregator devices require the support of cryptography operations on leaf IoT sensors, as well as energy-consuming protocol-level time synchronization strategies.

Without loss of generality, PPRQ can be adopted when there are strict constraints on the computational capabilities and the energy consumption of the IoT sensors and the aggregator. Moreover, PPRQ is suitable in any application where it is necessary to protect both the result of the MAX/MIN range query and the location and identity of the node providing the queried result. This is important in any application of MAX/MIN range queries where the IoT gateway needs to know the MAX/MIN value, but it does not need to know which specific IoT sensor originated the value, nor where it is located in the network. Overall, in scenarios where one of the above-discussed constraints is relaxed, other solutions currently available in the literature, such as [30],[21], and [32], can be adopted to efficiently manage privacy-preserving range query. However, PPRQ is intended to tackle a different scenario, where the constraints characterizing the IoT sensors and the aggregators make the integration of the above schemes neither suitable nor efficient.

Compared to all the solutions currently available in the literature, PPRQ is the first one that can achieve query accuracy, query privacy, and location privacy in MAX/MIN range query, without requiring the deployment of edge/fog nodes, and without forcing IoT sensors to support cryptography operations and protocol-level time synchronization protocols.

The only schemes that can provide either location privacy or query privacy by requiring low computational effort on aggregators and no cryptography support on IoT sensors are the schemes presented by the authors in [14], [15], and [16].

Therefore, to provide further insights, we compared the energy cost per device required by PPRQ and the proposals in [14], [15], and [16], taking as a reference the *single-message* configuration of our protocol, i.e., when PPRQ is set up with a single message for all the bits in the bit-string. Specifically, we assumed to integrate all the schemes in the IEEE 802.15.4 communication technology, which is one of the most popular MAC-layer technologies used in the context of IoT and Zigbee 3.0. For the energy analysis, we used the experimental data recently published in a publication of ours, assuming the use of the Openmote-b experimental hardware board [54]. The results of our investigation are reported in Figure 15, assuming the reference case of $K = 12$ bits, guaranteeing an error probability less than 0.01% for PPRQ, as detailed in Section VI-B. We also notice that we assume to deliver and receive packets of 127 bytes, i.e., the MTU of IEEE 802.15.4.

We notice that PPRQ is characterized by a limited and

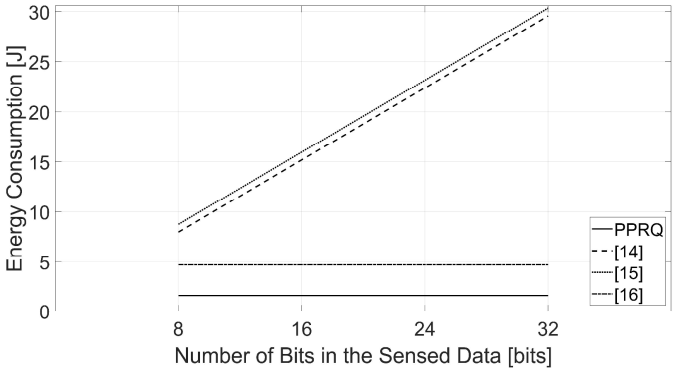

Fig. 15. Energy Consumption per device of PPRQ and the benchmark schemes in [14], [15], and [16], assuming an increasing size of the sensed value, in bits.

almost constant energy consumption per device, compared to the benchmark schemes in [14], [15], and [16]. Indeed, the proposal in [14] requires the delivery of a notification message to all the nodes for each round of the protocol and per each bit sensed by the leaf IoT devices, forcing all the nodes to have their radio up in RX mode. Similarly, the proposal by the authors in [15] requires the delivery of notification messages from the sink IoT nodes in every round of the protocol, for each bit, and a final message at the last round. All these operations are not required by PPRQ, since all the bits are delivered using a single IEEE 802.15.4 message. At the same time, the energy cost required by the proposal by the authors in [16] is reported by the authors in the reference publication, and it is related to a branching factor c, set to 5 by the reference authors. Therefore, for each round of the protocol, the proposal by the authors in [16] requires 5 reception operations and 1 transmission. Conversely, PPRQ requires only a single hash, a single reception (the query message from the sink IoT node), and a single transmission, allowing to reduce drastically the energy consumption. The energy consumption increases only a little with an increase of J, due to the transmission of more bits into the same MAC-layer packet. However, being the transmitted data always less than the MTU of 127 bytes, this overhead does not translate into additional RF messages, thus being almost constant.

Overall, considering the worst-case $J = 32$ bits, PPRQ requires only $1,581$ J, which is 33.67% less than the least energy-demanding competing solution, that is the one by the authors in [16]. We also emphasize that this comparison does not take into account the cost necessary to maintain protocol-level time synchronization between the nodes, as well as transmission from other nodes in the network, that are required in the benchmark schemes but not by our PPRQ scheme.

We also notice that the PPRQ protocol is not required to be always active in the network. Indeed, it can be triggered on request (by an IoT sensor node, based on the sensitivity of the data to be reported, or by the sink IoT node), and de-activated

TABLE II
QUALITATIVE COMPARISON OF PPRQ AGAINST STATE-OF-THE-ART APPROACHES FOR PRIVACY-PRESERVING RANGE QUERYING.

Ref.	No need to Support Cryptography Operations on IoT sensors	No need for Protocol-Level Time Synchronization	Computational and Storage Effort on Aggregators	Query Privacy	Location Privacy
[14]	✓	✗	Low	✗	✓
[15]	✓	✗	Low	✓	✗
[16]	✓	✗	Low	✓	✗
[17]	✗	✗	High	✓	✗
[19]	✗	✗	High	✓	✗
[20]	✗	✗	High	✓	✓
[21]	✗	✗	High	✓	✗
[22]	✗	✗	High	✓	✓
[24]	✗	✗	High	✓	✗
[25]	✓	✗	High	✓	✓
[26]	✗	✗	High	✓	✗
[27]	✗	✗	High	✓	✗
[28]	✗	✗	High	✓	✗
[29]	✗	✗	High	✗	✓
[30]	✗	✗	High	✓	✗
[32]	✗	✗	High	✓	✗
PPRQ	✓	✓	Low	✓	✓

when the reported data are not sensitive.
Therefore, PPRQ can guarantee both query privacy and location privacy, by requiring a low computational effort both on IoT sensors and aggregators. Further, it is completely tunable, and it also enjoys a high degree of flexibility, resulting in an overall energy-efficient solution.

VII. CONCLUSION

In this paper, we presented PPRQ, a lightweight and privacy-preserving protocol that provides data privacy and location privacy in IoT networks range querying. PPRQ achieves such ambitious objectives requiring just low-cost hashing and bitwise xor operations, both on intermediate (aggregator) nodes and on the sink IoT node. Therefore, compared to existing solutions, PPRQ is extremely lightweight, and it does not require the deployment of dedicated edge or fog nodes, and not even a mains-supplied computing unit, hence resulting applicable in a variety of use-cases.

Overall, the results reported in the paper demonstrate that PPRQ is extremely reliable. Indeed, it can be configured to guarantee a success probability higher than 99%, irrespective of the number of IoT sensors compromised by the adversary (provided it does not control the totality of IoT nodes), by just sending, for a wide of range of applications, just a single packet. In addition, PPRQ is also robust against an active adversary trying to disrupt the protocol computations. These performance are achieved through a careful choice of the replication parameter K, that trades-off resilience with a slight increase of the communication overhead.

Based on the above features, PPRQ emerges as an ideal solution for IoT networks range querying, especially when employed in remote locations, where dedicated edge and fog nodes cannot be conveniently deployed, and mains-supply is hardly available, such as remote sensing, exploration, agriculture, and military IoT applications.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their comments and insights, that have helped improving the quality of the paper.
This publication was partially supported by the award NPRP 11S-0109-180242 from the Qatar National Research Fund (QNRF), a member of The Qatar Foundation. The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the QNRF.

REFERENCES

[1] S. Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2101–2132, 2018.
[2] J. B. Hoffmann, P. Heimes, and S. Senel, "IoT Platforms for the Internet of Production," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4098–4105, June 2019.
[3] T. Yu, X. Wang, and A. Shami, "Recursive Principal Component Analysis-Based Data Outlier Detection and Sensor Data Aggregation in IoT Systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2207–2216, 2017.
[4] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," *Journal of Network and Computer Applications*, vol. 97, pp. 23 – 34, 2017.
[5] J. Qi, Y. Kim, C. Chen, X. Lu, and J. Wang, "Demand Response and Smart Buildings: A Survey of Control, Communication, and Cyber-Physical Security," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 4, October 2017.
[6] P. Tedeschi and S. Sciancalepore, "Edge and Fog Computing in Critical Infrastructures: Analysis, Security Threats, and Research Challenges," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*. IEEE, 2019, pp. 1–10.
[7] S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, and K. Choo, "Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things," *Future Generation Computer Systems*, vol. 91, pp. 382–391, 2019.
[8] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758–3773, 2018.
[9] K. Chan and F. T. Johnsen, "Military Communications and Networks," *IEEE Communications Magazine*, vol. 58, no. 2, pp. 60–60, 2020.
[10] B. Hamdaoui, N. Zorba, and A. Rayes, "Participatory IoT networks-on-demand for safe, reliable and responsive urban cities," *IEEE Blockchain Technical Briefs*, 2019.
[11] J. An, X. Gui, Z. Wang, J. Yang, and X. He, "A Crowdsourcing Assignment Model Based on Mobile Crowd Sensing in the Internet of Things," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 358–369, 2015.

[12] R. Roman, R. Rios, J. A. Onieva, and J. Lopez, "Immune System for the Internet of Things using Edge Technologies," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[13] H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan. 2018.

[14] R. Di Pietro and A. Viejo, "Location privacy and resilience in wireless sensor networks querying," *Computer Communications*, vol. 34, no. 3, pp. 515 – 523, 2011, special Issue of Computer Communications on Information and Future Communication Security.

[15] H. Dai, Y. Ji, F. Xiao, G. Yang, X. Yi, and L. Chen, "Privacy-Preserving MAX/MIN Query Processing for WSN -as-a-Service," in *IFIP Networking Conference (IFIP Networking)*, 2019, pp. 1–9.

[16] M. M. Groat, W. Hey, and S. Forrest, "KIPDA: k-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *Proceedings IEEE INFOCOM*, 2011, pp. 2024–2032.

[17] F. Chen and A. X. Liu, "Privacy- and Integrity-Preserving Range Queries in Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1774–1787, Dec 2012.

[18] L. Marconi, M. Conti, and R. Di Pietro, "CASSANDRA: a probabilistic, efficient, and privacy-preserving solution to compute set intersection," *Int. Journ. of Information Security*, vol. 10, no. 5, p. 301, 2011.

[19] Y. Yao, N. Xiong, J. H. Park, L. Ma, and J. Liu, "Privacy-preserving max/min query in two-tiered wireless sensor networks," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1318 – 1325, 2013.

[20] Q. Kong, R. Lu, M. Ma, and H. Bao, "A Privacy-Preserving and Verifiable Querying Scheme in Vehicular Fog Data Dissemination," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1877–1887, Feb 2019.

[21] R. Lu, "A New Communication-Efficient Privacy-Preserving Range Query Scheme in Fog-Enhanced IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2497–2505, 2019.

[22] 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*, vol. 25, no. 4, pp. 2334–2346, Aug. 2017.

[23] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in Internet of Things with Privacy Preserving: Challenges, Solutions and Opportunities," *IEEE Network*, vol. 32, no. 6, pp. 144–151, Nov. 2018.

[24] 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 *GLOBECOM 2017*, Dec. 2017, pp. 1–7.

[25] X. Zhang, L. Dong, H. Peng, H. Chen, S. Zhao, and C. Li, "Collusion-Aware Privacy-Preserving Range Query in Tiered Wireless Sensor Networks," *Sensors*, vol. 14, no. 12, pp. 23 905–23 932, 2014.

[26] B. Sheng and Q. Li, "Verifiable Privacy-Preserving Range Query in Two-Tiered Sensor Networks," in *IEEE INFOCOM*, 2008, pp. 46–50.

[27] B. Sheng and Q. Li, "Verifiable Privacy-Preserving Sensor Network Storage for Range Query," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1312–1326, 2011.

[28] 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*, vol. 2016, 2016.

[29] H. Dai, M. Wang, X. Yi, G. Yang, and J. Bao, "Secure MAX/MIN Queries in Two-Tiered Wireless Sensor Networks," *IEEE Access*, vol. 5, pp. 14 478–14 489, 2017.

[30] H. Mahdikhani, R. Lu, Y. Zheng, J. Shao, and A. A. Ghorbani, "Achieving $O(log^3 n)$ Communication-Efficient Privacy-Preserving Range Query in Fog-Based IoT," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5220–5232, 2020.

[31] M. Wen, R. Lu, K. Zhang, J. Lei, X. Liang, and X. Shen, "PaRQ: A Privacy-Preserving Range Query Scheme Over Encrypted Metering Data for Smart Grid," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 178–191, 2013.

[32] H. Bao and L. Chen, "A lightweight privacy-preserving scheme with data integrity for smart grid communications," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1094–1110, 2016.

[33] R. Li, A. X. Liu, A. L. Wang, and B. Bruhadeshwar, "Fast range query processing with strong privacy protection for cloud computing," *Proc. of the VLDB Endowment*, vol. 7, no. 14, pp. 1953–1964, 2014.

[34] R. Li and A. X. Liu, "Adaptively Secure Conjunctive Query Processing over Encrypted Data for Cloud Computing," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, 2017, pp. 697–708.

[35] L. Marconi, R. Di Pietro, B. Crispo, and M. Conti, "Time Warp: How Time Affects Privacy in LBSs," in *Int. Conf. on Information and Communications Security*. Springer, 2010, pp. 325–339.

[36] S. Ortolani, M. Conti, B. Crispo, and R. Di Pietro, "Events privacy in WSNs: A new model and its application," in *IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks*, 2011, pp. 1–9.

[37] M. Conti, J. Willemsen, and B. Crispo, "Providing Source Location Privacy in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1238–1280, Third 2013.

[38] G. Sun, V. Chang, M. Ramachandran, Z. Sun, G. Li, H. Yu, and D. Liao, "Efficient location privacy algorithm for Internet of Things (IoT) services and applications," *Journal of Network and Computer Applications*, vol. 89, pp. 3–13, 2017.

[39] G. Sun, D. Liao, H. Li, H. Yu, and V. Chang, "L2P2: A location-label based approach for privacy preserving in LBS," *Future Generation Computer Systems*, vol. 74, pp. 375–384, 2017.

[40] C. Yin, J. Xi, R. Sun, and J. Wang, "Location Privacy Protection Based on Differential Privacy Strategy for Big Data in Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3628–3636, Aug 2018.

[41] S. Sciancalepore, G. Piro, E. Vogli, G. Boggia, L. A. Grieco, and G. Cavone, "LICITUS: A lightweight and standard compatible framework for securing layer-2 communications in the IoT," *Computer Networks*, vol. 108, pp. 66–77, 2016.

[42] K. Yang, D. Blaauw, and D. Sylvester, "Hardware Designs for Security in Ultra-Low-Power IoT Systems: An Overview and Survey," *IEEE Micro*, vol. 37, no. 6, pp. 72–89, Nov. 2017.

[43] S. Sciancalepore, G. Oligeri, G. Piro, G. Boggia, and R. Di Pietro, "EXCHANge: Securing IoT via channel anonymity," *Computer Communications*, vol. 134, pp. 14–29, 2019.

[44] R. Di Pietro and G. Oligeri, "COKE crypto-less over-the-air key establishment," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 163–173, 2012.

[45] A. K. Das, A. Tabassum, S. Sadaf, and D. Sinha, "Attack Prevention Scheme for Privacy Preservation (APSP) Using K Anonymity in Location Based Services for IoT," in *Computational Intelligence in Pattern Recognition*. Springer, 2020, pp. 267–277.

[46] A. K. Das, S. Kalam, N. Sahar, and D. Sinha, "UCFL: User Categorization using Fuzzy Logic towards PUF based Two-Phase Authentication of Fog assisted IoT devices," *Computers & Security*, p. 101938, 2020.

[47] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. Culler, "SPINS: Security protocols for sensor networks," *Wireless networks*, vol. 8, no. 5, pp. 521–534, 2002.

[48] W. Stallings, *Cryptography and network security: principles and practice*. Pearson Upper Saddle River, 2017.

[49] C. Lesjak, D. Hein, and J. Winter, "Hardware-security technologies for industrial IoT: TrustZone and security controller," in *IECON 2015*, Nov 2015, pp. 2589–2595.

[50] S. Berger, R. Cáceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn, "vTPM: Virtualizing the Trusted Platform Module," in *Proc. USENIX Security Symp.*, ser. USENIX-SS'06, 2006.

[51] E. De Cristofaro and R. Di Pietro, "Adversaries and Countermeasures in Privacy-Enhanced Urban Sensing Systems," *IEEE Systems Journal*, vol. 7, no. 2, pp. 311–322, 2013.

[52] L. Zhang, H. Zhang, M. Conti, R. Di Pietro, S. Jajodia, and L. V. Mancini, "Preserving privacy against external and internal threats in WSN data aggregation," *Telecommunication Systems*, vol. 52, no. 4, pp. 2163–2176, 2013.

[53] S. Sciancalepore, G. Oligeri, and R. Di Pietro, "Strength of Crowd (SOC) - Defeating a Reactive Jammer in IoT with Decoy Messages," *Sensors*, vol. 18, no. 10, p. 3492, 2018.

[54] P. Tedeschi, S. Sciancalepore, A. Eliyan, and R. Di Pietro, "LiKe: Lightweight Certificateless Key Agreement for Secure IoT Communications," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 621–638, 2020.

Savio Sciancalepore is currently Post Doc at HBKU-CSE-ICT, Doha, Qatar. He received his master degree in Telecommunications Engineering in 2013 and the PhD in 2017 in Electric and Information Engineering, both from the Politecnico di Bari, Italy. He received the prestigious award from the ERCIM Security, Trust, and Management (STM) Working Group for the best Ph.D. Thesis in Information and Network Security in 2018. His major research interests include network security issues in Internet of Things (IoT) systems and Cyber-Physical Systems, including UAV networks, avionics systems, and mobile networks.

Roberto Di Pietro, ACM Distinguished Scientist, is Full Professor in Cybersecurity at HBKU-CSE. Previously, he was in the capacity of Global Head Security Research at Nokia Bell Labs, and Associate Professor (with tenure) of Computer Science at University of Padova, Italy. He also served 10+ years as senior military technical officer. Overall, he has been working in the cybersecurity field for 23+ years, leading both technology-oriented and research-focused teams in the private sector, government, and academia (MoD, United Nations HQ, EUROJUST, IAEA, WIPO). His main research interests include security and privacy for wired and wireless distributed systems (e.g. Blockchain technology, Cloud, IoT, On-line Social Networks), virtualization security, applied cryptography, computer forensics, and data science. Other than being involved in M&A of start-up—and having founded one (exited)—, he has been producing 230+ scientific papers and patents over the cited topics, has co-authored three books, edited one, and contributed to a few others. He is serving as an AE for ComCom, ComNet, PerCom, Journal of Computer Security, and other Intl. journals. In 2011-2012 he was awarded a Chair of Excellence from University Carlos III, Madrid. In 2020 he received the Jean-Claude Laprie Award for having significantly influenced the theory and practice of Dependable Computing.