# k-Anonymous Query Scheme on the Internet of Things: a Zero Trust Architecture

Kadhim Hayawi[1], Pin-Han Ho[2], Sujith Samuel Mathew[1], and Limei Peng[3]
[1]College of Technological Innovation, Zayed University, Abu Dhabi, UAE
[2]Department of Electrical and Computer Engineering, University of Waterloo, Canada
[3]Department of Industry Engineering, Ajou University, Korea

**The paper investigates query-anonymity in Internet of things (IoT) formed by a sensor cloud, where the sensor nodes provide services of sensing and are subject to user queries of sensing data. Due to the heterogeneity and multi-carrier natures of the sensor cloud, user privacy could be impaired when the queries have to go through nodes of a third party. Thus, the paper firstly introduces a novel query k-anonymity scheme that countermeasures such a privacy threat. Based on the proposed k-anonymity scheme, the trade-offs between the achieved query-anonymity and various performance measures including, communication-cost, return-on-investment metric, path-length, and location anonymity metrics, are analyzed. By adopting a hybrid approach that takes into account the average and worst-case analysis, our evaluation results show that most of the obtained bounds on various performance anonymity trade-offs can be expressed precisely in terms of the offered level-of-anonymity $k$ and network diameter $d$.**

***Index Terms*—Anonymity Performance Trade-offs, Cloud, Internet of Things, k-Anonymity, Privacy Preserving Protocols, Zero Trust Architecture.**

## I. Introduction

The Internet of Things (IoT) introduces a paradigm shift in networking that aims to connect real-world things to the Internet. These are things that collect information with potential value for information consumers. These IoT candidates are embedded with computing capabilities, communication protocols, and sensors to blend into everyday environments and seamlessly bridge the gap between people, places, and things over the Internet [1].

Although many real-world things have valuable information to share, it is a challenge in equipping them with all the required capabilities to connect to the Internet. The wide adoption and enhancement of technologies like RFID, cloud computing, and Wireless Sensor Networks (WSN) have enabled the virtualization of the real-world things [1], [2]. For instance, things with embedded sensors provide sensory data that are accessed as Web services in the cloud and create the possibility of composing different services to infer important knowledge about physical environments [2]. The sensed knowledge is used to enhance the capability of the things that do not have sensing capability but are in the same location as those things that have sensing capabilities. This shift in paradigm has opened a plethora of potential smart applications via the provision of sensing as a service (SaaS) [2], [3]. Today, the control and management of thousands of these things via cloud services is a norm, where real-world things are virtualized and further enhanced in their capabilities and semantics [4], [5].

It is clear that IoT matures to provide flexible, scalable, and real-time communications with the physical world in a ubiquitous way, which nonetheless leads to security and privacy concerns at the same moment. The trust factor of the clients could be significantly damaged in the event that the client queries upon a physical sensor are actually received and handled by a third-party entity (e.g., an untrusted cloud service provider). As shown in Fig. 1, the owners of the physical sensors and the sensor cloud are decoupled from the client via one or multiple virtual sensors. The exposure of real-world environments to the virtual world requires the IoT candidates to have the option of being anonymous. In this case, a query-anonymity scheme is essential to mitigate the privacy concern, and its security, anonymity, and overhead should be well defined and quantified.

Manuscript received July 6, 2021; revised November 16, 2021. Corresponding author: Pin-Han Ho (email: p4ho@uwaterloo.ca).

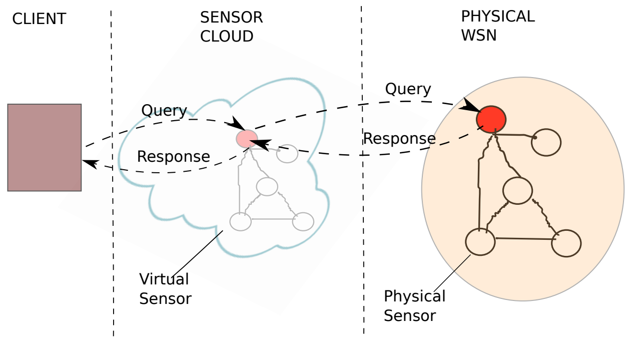


Fig. 1: The client queries the sensors in the sensor cloud and hopes the destination sensor node to be anonymous.

To address the above mentioned problem, the study considers an application scenario in which the client is concerned about the privacy of the destination of his query. We take a communication-based approach where the client is required to send a query message to additional $k-1$ destinations in order to hide its destination of interest. These $k$ destinations form an anonymity-set [6]–[10] that prevents any untrusted party from knowing which destination is of interest to the client. Certainly such a redundancy causes additional communication cost as a trade-off [11], [12]. As the most important endeavour of the paper, we adopt an experimental approach to analyze these trade-offs in detail to study the average and worst-case bounds on them, and investigate several variations of anonymity-sets constructions methods. We are committed to answer the following questions: what is the impact of these variations on the performance of secure query $k-anonymity$ scheme with respect to identity and location-anonymity? Which of these extensions are worthwhile in terms of the cost-effective secure query $k-anonymity$ solutions they offer?

The contributions of this paper are in two folds. Firstly, we propose a novel secure query k-anonymity scheme and introduce its detailed implementation, including the partition algorithm, anonymity-sets construction methods, query routing algorithm, and querying protocol (see section IV). The main design goal is to lower the incurred communication-cost while satisfying the desired level-of-anonymity $k$. Secondly, we analyze the performance of the proposed scheme in depth by looking into several performance metrics in various simulation settings (see section V). During the analysis, we embrace a hybrid approach that takes both the average and worst-case scenarios into consideration. All the targeted variables, such as communication-cost, cost-benefit metric, path length, and location-anonymity metrics are measured for large values of independent variables namely, level-of-anonymity $k$, network size $n$, and network diameter $d$. Our evaluation results establish bounds on various anonymity performance trade-offs of the secure k-anonymous query scheme in terms of their offered level-of-anonymity $k$, and network diameter $d$.

## II. Background and terminology

The section presents the trust model, network model, and threat model of the study.

### A. Zero Trust Model

We consider a Large WSN conglomerate owned by multiple operators, and the WSN could be in any type like terrestrial, underground, underwater, and mobile networks for gathering environmental data. The clients query the physical sensors for the sensed data by accessing services provided by the sensor-cloud providers which could be governments, corporates, or academia. The various stakeholder entities in the scenario include the operators who own the WSNs, the cloud service providers, and the clients that query the sensors. Their relation is that the cloud service providers lease resources from the WSN operator/owner to offer the SaaS to their clients [2]. Thus, it is the cloud service providers who face their clients but are with limited knowledge on the sensor-cloud operations. Many applications of this model are foreseen in areas like smart cities, smart transportation, and smart manufacturing. The clients use the sensor data for different purposes like traffic management, resource exploration, environmental protection, air pollution control, and civil infrastructure monitoring, etc.

With the above mentioned scenario, we consider the *trust-none model* where a client trusts no other entity [11], which has been termed recently as ”zero trust model” [13]–[16]. The clients are concerned about the privacy of their interest, queries, and data access patterns which may be compromised because of untrusted sensor-cloud owners or other competing clients.

An interesting application scenario is the deep-sea exploration in oil fields, where an oil company (client) is interested in querying the deep sea WSN that is owned by operators that are different from the cloud service provider [12]. Certainly the oil company (client) is not willing to share its interest with other stakeholder entities. It follows that a zero Trust model is such that the client treats the WSN, the cloud service provider, and other clients as untrustworthy adversaries. The k-anonymous query scheme provides receiver anonymity service by hiding the ID of the destination node of a query in the crowd of other $k-1$ queries' destinations. This is achieved by querying additional $k-1$ nodes along with the original query, such that an adversary cannot learn the true destination of a k-anonymous query with a probability non-negligibly greater than $1/k$ by analyzing its traffic patterns. To satisfy this and to defend against all types of traffic analysis attacks including the intersection and statistical attacks, the proposed scheme divides the network of size $n$ into multiple disjoint anonymity-sets of size $k \in [1, n]$.”

### B. WSN Network Model

Since query-anonymity solutions in a large-scale WSN are considered, we group the WSN sensor nodes into clusters whose head are elected or pre-assigned by network designer [17]. Thus the WSN is modeled as undirected connected graph $G(V, E)$, where $V$ (vertices) are a number of $n$ cluster heads joined by links or edges $E$. For every two nodes $v_i, v_j \in V$, it holds that the edge $(v_i, v_j) \in E$ if and only if whatever $v_i$ transmits, $v_j$ can always receive, i.e., $v_i$ is adjacent to $v_j$. It is important not to confuse WSN sensor nodes with the nodes in the graph, where the nodes in the graph are cluster-heads that are richer in resources compared to their cluster members and serve as gateway for relaying/broadcasting the clients' queries. We consider all cluster heads to be equal in power, computation, and communication capabilities.

### C. Threat Model

The adversary in our setting is an unconditional global passive eavesdropper who is able to monitor and analyze all the message exchange between the client and the WSN but does not actively alter them. This is an often adopted adversarial model called the *semi-honest* or *honest-but-curious* adversary [18]. The adversary is global and strong enough to have access to the entire WSN. When there is no limitation on the computational power of the adversary or on her ability to collude with other component of the system, it is unconditional. The adversary is able to conduct cipher-text only traffic analysis attack. The attacker has access only to the output of the anonymity transformation when executed.

## III. Problem formulation

In this section, the problem of designing a secure k-anonymous query scheme is formulated. Based on the DAS (Disjoint Anonymity-Sets) construction [19], the following three sub-tasks should be defined: one partition algorithm $\pi$ and two anonymity transformations $T$ and $T^{-1}$.

1) $\pi$ is the partition algorithm that converts the whole WSN of size $n$ into a union of disjoint non-empty anonymity-sets. Specifically, the inputs of the algorithm include $n$ the size of WSN and $k \in [1, n]$ the desired level-of-anonymity. It outputs a partition of WSN containing a set disjoint anonymity-sets $\mathbb{S} = \{s_1, \ldots, s_{\lfloor n/k \rfloor}\}$, where $\cup_{s \in \mathbb{S}} = WSN$, and each $s \in S$ is of size $|s| = [k, 2k-1]$. The upper limit point of anonymity-set size, which is $2k-1$, captures that fact that $n$ may not be divisible by $k$. Hence, there exists at least one anonymity-set of size in the closed interval $[k, 2k-1]$.
2) $T$ is the anonymity transformation that maps every query $q$ in its input space $Q_T \in \mathbb{Q}_{\mathbb{T}}$ to the corresponding anonymity-set $s \in \mathbb{S}$, i.e., $\forall q \in Q_T, \exists s = T(q)$, where $q \in s$. This means whenever the client wants to send a query $q$ to a destination node, she is required to query each node in its corresponding anonymity-set $s$, as depicted in Fig. 2 for the case of $|s| = 3$. Moreover $s = Q_T$, and thus the partition algorithm $\pi$ defines not only the set of anonymity-sets $\mathbb{S}$ but also the set of input spaces of anonymity transformation $\mathbb{Q}_{\mathbb{T}}$.
Since the DAS anonymity-sets are disjoint, the input spaces of $T$ are disjoint too, which by nature resists all types of intersection attacks among different anonymity-sets [20]–[22]. This assures a certain level-of-anonymity regardless of the computational power of the eavesdropping adversary who is able to intercept the anonymity scheme for sufficiently long time.
3) $T^{-1}$ is the inverse transformation that recovers the true-destination query from anonymity-set of the responses to the queries generated by anonymity transformation $T$. That is $s = T^{-1}(q)$. Note that $T$ and $T^{-1}$ can both be performed by the client which meets the requirement of trust-none model.

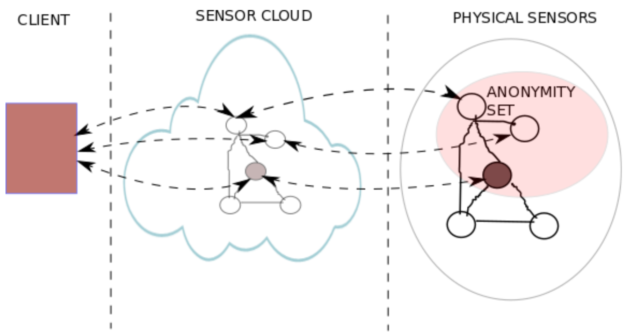


Fig. 2: The client queries additional nodes to hide its true destination (shaded).

## IV. Proposed SAS Scheme

Detailed implementation to each of the sub-tasks under DAS is provided in this section.

### A. The Partition Algorithm

We consider a source-routed square grid topology that has been widely employed in the related research [12], where the position of each node is defined by the ordered pair of its Cartesian coordinates $(i, j)$, where $i, j \in [1, \sqrt{n}]$. As in prior work [12], the client communicates directly with root-node $v_{1,1}$, which is the node at the upper left corner. The route between the root-node and any node in the shortest Manhattan distance. Note that Manhattan distance $d_m$ between two nodes $v_{i_1,j_1}$ and $v_{i_2,j_2}$ is equal to the length of the path connecting them along the horizontal and vertical segments [23].

The following theorem formally defines the proposed partition algorithm and its proof justifies its feasibility.

*Theorem 1:*
Given a $\sqrt{n} \times \sqrt{n}$ square grid connected undirected graph $G = (V, E)$, there exists a partition of $V$ into disjoint sets of nodes $\mathbb{S} = \{s_1, s_2, \ldots, s_{\lfloor n/k \rfloor}\}$ each of size $\in [k, 2k-1]$, the DAS anonymity-sets, such that the following properties hold:

1) The set of nodes $\{v_1, v_2, \ldots, v_{|s_j|}\}$ in such anonymity-set $s_j$ has the following total ordering. Given $i \in [1, |s_j|-1]$, it holds that the Manhattan distance between nodes $v_i$ and $v_{i+1}$ is exactly 1.
2) $v_1$ (the first-node from the source along the route) is at the lowest vertical distance (y-coordinate) from the root-node $v_{1,1}$ compared to other nodes in such anonymity-set $s_j$.
3) Nodes $\{v_2, v_3, \ldots, v_{|s_j|}\}$ in such anonymity-set $s_j$ is at a vertical distance of $\lceil |s|/\sqrt{n} \rceil$ from $v_1$.
4) The remainder out of partition $n/k$ is distributed over anonymity-sets $\mathbb{S}$ using either First-Set-Spread (FSS), Equal-Spread (ES), or Random-Spread (RS) anonymity-sets construction methods (which will be defined right below).

The properties (1) - (3) in the theorem ensure that the communication-cost is minimized when querying multiple sensor nodes in an anonymity-set, by which $v_1$ (the first-node along the route) is at the least vertical distance from $v_{1,1}$, while all the other nodes of the same anonymity-set is confined within a vertical distance of $\lceil |s|/\sqrt{n} \rceil$ from $v_1$.

The property (4) in the theorem ensures that the remainder nodes out of the partition $n/k$ is distributed to the anonymity-sets by using one of the following three anonymity-sets construction methods. The first is called First-Set-Spread (FSS), which adds all the remaining nodes to the first set closet to the root-node in order to lower the communication-cost because the first set is the closet to the root-node. The second is called Equal-Spread (ES), which goes to the other extreme by distributing the remainder equally over all the anonymity-sets. Whereas the Random-Spread (RS) adopts the uniform random distribution to disseminate the remaining nodes randomly across anonymity-sets. The three methods will be examined in Section V.

For the proof of existence of such disjoint anonymity-sets, our approach is to introduce a partition algorithm that can be implemented in any 2-connected planner networks and the resultant a partition meets all the properties listed in theorem 1. The algorithm calls four other procedures, namely $ConstructSpanningTree$ that constructs a comb-like spanning tree $t$ of $G$, $ConstructHamiltonianPath$ that constructs the Hamiltonian path $P$ of $t$, $FindAnonymitySetsSizes$ that determines the size of each anonymity-set, and $ConstructOneAnonymitySet$ that constructs one anonymity-set. A working example of the algorithm that uses First-Set-Spread (FSS) is shown in Fig. 3.

**Algorithm 1** Partition Algorithm ($\pi$)

```
Input: $G = (V, E)$: $\sqrt{n} \times \sqrt{n}$ Square grid connected undirected graph , $k$: a desired level of anonymity, and $ascm$: an anonymity-sets construction method
Output: $\mathbb{S}$: the partition of $G$ into disjoint anonymity-sets
1: $n \leftarrow |V|$
2: $t \leftarrow ConstructSpanningTree(G)$
3: $P \leftarrow ConstructHamiltonianPath(t)$
4: $\{|s_i|\,|i \in [1, \lfloor n/k \rfloor]\} \leftarrow FindAnonymitySetsSizes(ascm, n, k)$
5: foreach $i \in [1, \lfloor n/k \rfloor]$ do
6:     $s_i \leftarrow ConstructOneAnonymitySet(P, |s_i|)$
7:     $P \leftarrow P$ with all nodes of $s_i$ removed
8: end for
9: return $\mathbb{S} \leftarrow \{s_i | i \in [1, \lfloor n/k \rfloor]\}$
10: procedure $ConstructSpanningTree(G')$
    /* BFS is the Breadth First Search algorithm */
11: return $t' \leftarrow BFS(G')$ starting at root-node $v_{1,1}$
12: procedure $ConstructHamiltonianPath(t')$
    /* $t'$ is a comb-like tree constructed by $ConstructSpanningTree$ procedure above */
13: for $i \leftarrow 1$ to $\sqrt{n}$ do
14:     if $i$ is odd then
15:         Remove the edge between nodes $(i, 1)$ and $(i + 1, 1)$ in $t'$
16:         Insert an edge between nodes $(i, \sqrt{n})$ and $(i + 1, \sqrt{n})$ in $t'$
17:     end if
18: end for
19: return $P' \leftarrow t'$
20: procedure $FindAnonymitySetsSizes(ascm', n', k')$
21: foreach $i \in [1, \lfloor n'/k' \rfloor]$ do
22:     $|s'_i| \leftarrow k'$
23: end for
24: $r \leftarrow n'$ mod $k'$
25: if $ascm = FSS$ then
26:     $|s'_1| \leftarrow |s'_1| + r$
27: else if $ascm = ES$ then
28:     foreach $j \in [1, r]$ do
29:         $x \leftarrow j$ mod $\lfloor n'/k' \rfloor$
30:         if $x = 0$ then
31:             $x \leftarrow \lfloor n'/k' \rfloor$
32:         end if
33:         $|s'_x| \leftarrow |s'_x| + 1$
34:     end for
35: else if $ascm = RS$ then
36:     foreach $j \in [1, r]$ do
37:         $x \leftarrow$ RandomNumberGenerator(1, $\lfloor n'/k' \rfloor$)
38:         $|s'_x| \leftarrow |s'_x| + 1$
39:     end for
40: end if
41: return $\{|s'_i|\,|i \in [1, \lfloor n'/k' \rfloor]\}$
42: procedure $ConstructOneAnonymitySet(P', l)$
43: if $l >$ number of nodes in $P'$ then
44:     return error
45: end if
    /* $P'$ is a Hamiltonian path that we treat as a tree. */
46: Pre-order traverse $P'$ /* traverse from the the root-node, $v_{1,1}$ */
47: return a set containing the first $l$ nodes we encounter
```

The Partition algorithm takes as inputs a $\sqrt{n} \times \sqrt{n}$ square grid connected undirected graph $G = (V, E)$, a desired level of anonymity $k \in [1, n]$, and an anonymity-sets construction method (ascm). In line 2, we call $ConstructSpanningTree$ to generate a comb-like spanning tree $t$ of $G$. Then, we invoke $ConstructHamiltonianPath$ on $t$ in line 3.

In $ConstructSpanningTree$, line 10, we invoke Breadth First Search (BFS) [24], [25] at root-node, $v_{1,1}$, which visits all of its adjacent nodes from left to right. Then for each of these adjacent nodes in turn, BFS visits their adjacent nodes from left to right if they are unvisited before, and continues in a similar manner. This forms a comb-like spanning tree as shown in Fig. 4 that has one spine of length $\sqrt{n}$ at $y = 1$ vertical line, and $\sqrt{n}$ teeth, each of length $\sqrt{n}$, along the $x$ axis.

In $ConstructHamiltonianPath$, line 12, we convert the comb-like spanning tree $t$ into a Hamiltonian path $P$. We accomplish this by removing some edges and replacing them with others in different locations. For doing this, In line 13, we iterate $i$ through the range of rows in the square grid. That is, it takes on values from 1 through $\sqrt{n}$. In line 14, we limit $i$ values to odd numbers. In the $i^{th}$ iteration, we remove the edge between each two consecutive rows ($i$ and $i+1$) for odd values of $i$, and join them by an edge on the last column, as in lines 15 and 16.

To see why, refer to the comb-like tree $t$ shown in the second graph from left of Fig. 3. To convert $t$ into a Hamiltonian path, we need to: remove the edges joining the first and second rows on the first column, join them by inserting an edge on the last column, $\sqrt{n}^{th}$, leave the edge between the second and the third rows intact, and repeat the remove and insertion between the third and fourth rows. This is done in two iterations of the **for** loop in line 13, as indicated on the resultant Hamiltonian path in Fig. 3.

Back to the main body of Algorithm 1, we determine the size of each anonymity-set by calling $FindAnonymitySetsSizes$ in line 4. This step is a prelude to assigning nodes to these anonymity-sets which is done in lines 5 to 8. In $FindAnonymitySetsSizes$, we set up an iterator, $i$, through all the anonymity-sets which takes values from 1 till $\lfloor n/k \rfloor$, line 21. In each iteration, we initialize the $i^{th}$ anonymity-set, $|s'_i|$, to be equal to the desired level-of-anonymity, $k'$. This is the minimum size of an anonymity-set since the anonymity-set size is in $[k, 2k - 1]$. In line 24, we calculate the remainder $r$ out of the partition $n/k$ using modulo operation.

Then, starting at line 25, we use **if-else if** statement to stress the fact that either one of the three anonymity-sets construction method (ascm), FSS, EQ, or RS, is used. As mentioned in the proceeding paragraphs, these methods are used to distribute the remainder $r$ over anonymity-sets. Simply, the FSS adds all the remaining nodes $r$ to the first anonymity-set in line 26. Recall that ES attempts to add equal share of the remaining nodes $r$ to each anonymity-set. Thus, we first set up an iterator, $j$, which takes values from 1 till the remainder $r$, line 28. In each iteration, we select an anonymity-set indexed by $x$ sequentially, lines 29 to 32. Then, we increment its size by one node from the remaining nodes in line 33. Finally, for the RS, we again set up an iterator, $j$, which takes values from 1 till the remainder $r$, line 36. In each iteration, we increase the size of an anonymity-set whose index $x$ is selected by a random number generator following the uniform distribution, lines 37 and 38.

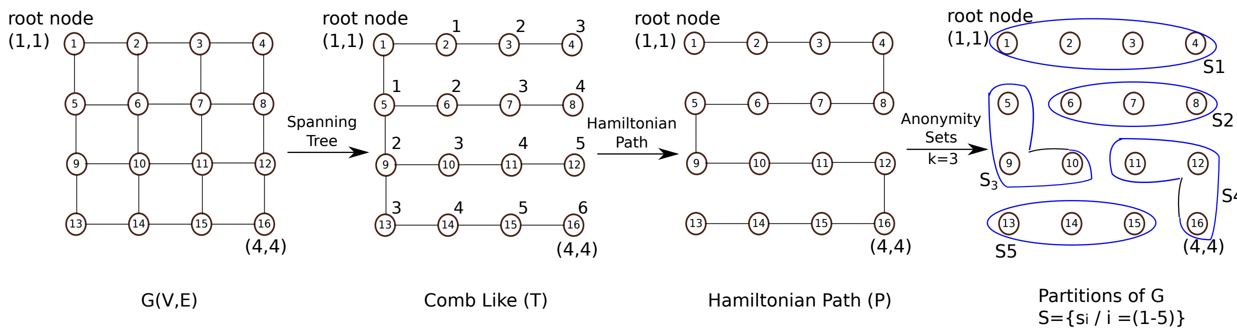


Fig. 3: An example of running the partition algorithm shown in Algorithm 1 on $4 \times 4$ square grid connected undirected graph .

In line 5 of the main body of the partition algorithm, we set up an iterator, $i$, through all the anonymity-sets which takes values from 1 till $\lfloor n/k \rfloor$. In each iteration, we call $ConstructOneAnonymitySet$, line 6, to construct the $i^{th}$ anonymity-set. Then, we update the Hamiltonian path by removing the nodes of constructed $i^{th}$ anonymity-set from it, as in line 7. Finally, after finishing the **for** loop in line 9, we return the partition $\mathbb{S}$ of $G$, which is a collection of anonymity-sets.

In $ConstructOneAnonymitySet$, we treat the Hamiltonian path $p'$, which is one of the inputs in addition to an integer $l$, as a simple example of a tree. In line 46, we traverse the tree in a pre-order way [26]. That is, we recursively visit parent node from left to right before visiting their children node. In essence, this means we walk down the Hamiltonian path $P'$ from its endpoint that is the root-node. In the last line, 47, we return the first $l$ nodes we encounter to create one anonymity-set.

An important assertion is that the path from the root-node to any other vertex $v$ in $t$ is the shortest path from root-node to $v$ in $G$. This can be easily verified via the fact that each connected graph has a spanning tree [25], as well as the fact that the shortest path spanning tree $t$ can be built via Breadth First Search (BFS) [24], [25] which we include in the procedure $ConstructSpanningTree$ of the algorithm. Respectively, we derive the Hamiltonian path from the same spanning tree that we will use later for routing queries to the anonymity-sets, and collecting their responses, see section IV-B.

Note that the anonymity-sets are contiguous since all the nodes are possibly queried. Thus in the algorithm, we generate these anonymity-sets as a result of splitting a continuous path originating at the root-node to ensure the continuity requirement. On the other hand, since the path must be a Hamiltonian path that visits every node in WSN once, the disjointedness requirement is naturally met.

**Time Complexity**: We show that time of running Algorithm 1 on $G = (V, E)$ is linear to the number of nodes $|V| = n$. First, the Initialization in line 1 takes $O(1)$ time. $ConstructSpanningTree$ takes $O(|V| + |E|)$ with the BFS algorithm [26]. Now, in a $\sqrt{n} \times \sqrt{n}$ square grid graph, we observe that $|V| = n$, and $|E| = 2(n - \sqrt{n})$. Hence $ConstructSpanningTree$ takes $O(|V|+|E|) = O(n+2(n-\sqrt{n})) = O(n)$.

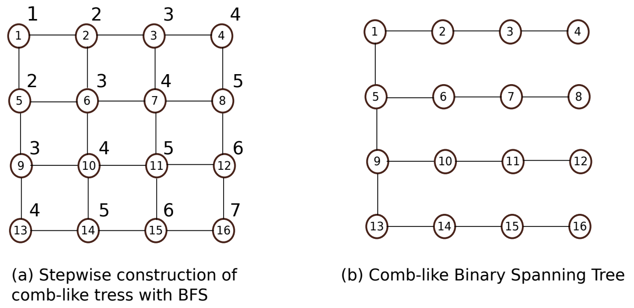

(a) Stepwise construction of comb-like tress with BFS
(b) Comb-like Binary Spanning Tree

Fig. 4: An example of comb-like rooted binary spanning tree construction algorithm using BFS for $4 \times 4$ square grid connected undirected graph. (a) Stepwise construction of the tree in which the sequence of steps is numbered on the diagram, (b) the resultant tree

In $ConstructHamiltonianPath$, the **for** loop costs $O(\sqrt{n})$ due to a constant time consumed by the execution of the body of the loop. $FindAnonymitySetsSizes$ in line 4 yields $O(n)$ of running time. This is so because its cost is mainly of two parts namely, the **foreach** loop and the **if-else if** statement, and each one of them is in $O(n)$ as follows. The **foreach** in line 21 costs $\lfloor n/k \rfloor$ since its body takes constant time. But $k \in [1, n]$, therefore **foreach** is in $O(n)$. The FSS case, line 25, of **if-else if** costs a constant time, i.e $O(1)$. The ES case costs the same of its **foreach** loop in line 28 which performs a constant time block $r$ times. Recall that $r$, the remainder out of the devision $n/k$, is in $O(k)$ which is in turn in $O(n)$. Hence the EQ case is in $O(n)$. Finally, the RS case is in $O(n)$ because its **foreach** performs a constant time block of code $r$ times. The **foreach** loop in lines 5–8 performs $ConstructOneAnonymitySet$ and the remove of $s_i$ nodes operation, line 7, $\lfloor n/k \rfloor$ times. Considering $ConstructOneAnonymitySet$, lines 42–45 costs a constant time, and the Pre-order, line 46, traverse only $|s_i|$ nodes of the Hamiltonian path. The remove operation, line 7, removes $|s_i|$ nodes from the Hamiltonian path. Since $|s_i| \in [k, 2k-1]$ (see section III), each of $ConstructOneAnonymitySet$ and the remove operation, line 7, takes time $\Theta(k)$. Therefore, we can state the running time of the **for** loop in lines 5–8 as $O(\lfloor n/k \rfloor \times (k + k) = O(n/k \times 2k) = O(n)$. Consequently, the total running time of the Partition algorithm, which is equal to the sum of the time taken by $ConstructSpanningTree$, $ConstructHamiltonianPath$, $FindAnonymitySetsSizes$, and the **foreach** loop in lines 5–8, is $O(n + \sqrt{n} + n + n) = O(n)$.

Now we are able to prove the existence of a valid partition according to theorem 1, which also proves the correctness of the Partition algorithm.

*proof 1 (Proof for Theorem 1):* Firstly we argue that by implementing the Partition Algorithm 1, the generated anonymity-sets are disjoint since each of them is obtained by splitting a Hamiltonian path of length $n$ into path segments of size in $[k, 2k-1]$. Then we argue that the four properties in theorem 1 are satisfied as follows. Property (1) is satisfied due to the fact that the Manhattan distance between nodes $v_i$ and $v_{i+1}$ is exactly 1, for $i \in [1, |s_j|-1]$, since they are successive nodes on the Hamiltonian path. To prove the feasibility of the second property, we specify $v_1$ of each anonymity-set to be the endpoint of the corresponding path segment that is closest to the root-node along the Hamiltonian path. The property is satisfied because the root-node, $v_{1,1}$, is at $y = 1$, the lowest y-coordinate. For example, $v_1$ becomes the root-node itself in the anonymity-set that contains the root-node, wherein the y-coordinate of $v_1$ is the lowest among other nodes in each anonymity-set, and nodes $\{v_2, v_3, \dots, v_{|s_j|}\}$ are located on the same row of $v_1$ or below it. Since the length of each row is $\sqrt{n}$, the number of rows that the anonymity-set spans is in $\Theta(\lceil |s|/\sqrt{n} \rceil)$, which satisfies the third property. The fourth property is naturally satisfied because in Algorithm 1, the remainder nodes are distributed over the anonymity-sets using one of the anonymity-sets construction methods namely, FSS, ES, or RS.

## B. Query Routing

To query a desired (by the client) destination node $v_d$ in WSN anonymously using DAS scheme, we need to route a query to every member of the anonymity-set $s_j$ to which $v_d$ belongs. The proposed routing algorithm achieves this by building a routing substrate based on the comb-like shortest-path spanning tree provided by the partitioning algorithm 1. By launching a single query that visits all nodes of $s_j$ to which $v_d$ belongs, the query responses of all the nodes are collected in a piggyback manner [27], where each node in the anonymity-set attaches its response data with its ID to the query packet.

The proposed source-route construction, namely Algorithm 2, is given as follows.

The output of Algorithm 2 is the list $h_j$ which contains the source-route header information required by the DAS anonymity scheme. Each entry in $h_j$ consists of two elements: the node ID value denoted by $ID_v$, and the node action value denoted by $Action_v$. As a query request is received, we first invoke $FindAnonymitySet$, in line 1, to find the the index $j \in [1, \lfloor n/k \rfloor]$ of the anonymity-set $s_j$ to which $v_d$ belongs. In $FindAnonymitySet$, line 10, an iterator $i$ is set up through all the anonymity-sets to check which one contains the queried node, $v_d$. In lines, 2 and 3, we specify $x$ and $y$ coordinates of the first-node as $v_1(x_1, y_1)$, and the last-node as $v_{|s_j|}(x_{|s_j|}, y_{|s_j|})$, of the discovered anonymity-set $s_j$. In line 4, we initialize $h_j$, which contains the source-route header information, to an empty list. In line 5, we call the procedure

**Algorithm 2** Source-Route Construction

```
Input: $G = (V, E)$: $\sqrt{n} \times \sqrt{n}$ connected undirected graph, $v_d \in V$: the
    queried destination node, and $\mathbb{S} = \{s_i | i \in [1, \lfloor n/k \rfloor]\}$: the anonymity-
    sets obtained from running Algorithm 1
Output: $h_j$: the source-route header information which is an ordered list of
    node IDs with their action values, and $s_j$: the corresponding anonymity-
    set of $v_d$, where $j \in [1, \lfloor n/k \rfloor]$
 1: $j \leftarrow FindAnonymitySet(v_d)$
    /* The coordinates of first-node of the anonymity-set $s_j$ */
 2: $(x_1, y_1) \leftarrow$ coordinates of $v_1$ of $s_j$
    /* The coordinates of last-node of the anonymity-set */
 3: $(x_{|s_j|}, y_{|s_j|}) \leftarrow$ coordinates of $v_{|s_j|}$ of $s_j$
 4: Initialize $h_j$ to an empty list
 5: $t \leftarrow ConstructSpanningTree(G)$
 6: $SourceRoutePart1(t, h_j, x_1, y_1)$
 7: $SourceRoutePart2(s_j, h_j)$
 8: $SourceRoutePart3(t, h_j, x_{|s_j|}, y_{|s_j|})$
 9: return $h_j, s_j$
10: procedure $FindAnonymitySet(v')$
11: for $i \leftarrow 1$ to $\lfloor n/k \rfloor$ do
12:     if $v' \in s_i$ then
13:         $j' = i$, break
14:     end if
15: end for
16: return $j'$
17: procedure $ConstructSpanningTree(G')$
    /* BFS is the Breadth First Search algorithm */
18: return $t' \leftarrow BFS(G')$ starting at root-node $v_{1,1}$
19: procedure $SourceRoutePart1(t', h'_j, x'_1, y'_1)$
20: for $i \leftarrow 1$ to $y'_1 - 1$ do
21:     $AppendForwardingNode((1, i), h'_j)$
22: end for
23: for $i \leftarrow 1$ to $(x'_1 - 1)$ do
24:     $AppendForwardingNode((i, y'_1), h'_j)$
25: end for
26: procedure $SourceRoutePart2(s'_j, h'_j)$
27: foreach node $v \in s'_j$ do
28:     $ID_v \leftarrow$ ID of node $v$
29:     $Action_v \leftarrow 1$ /* piggyback action */
30:     Append $\langle ID_v, Action_v \rangle$ to $h'_j$
31: end for
32: procedure $SourceRoutePart3(t', h'_j, x'_{|s_j|}, y'_{|s_j|})$
33: for $i \leftarrow x'_{|s_j|} - 1$ to 1 do
34:     $AppendForwardingNode((i, y'_{|s_j|}), h'_j)$
35: end for
36: for $i \leftarrow y'_{|s_j|} + 1$ to 1 do
37:     $AppendForwardingNode((1, i), h'_j)$
38: end for
39: procedure $AppendForwardingNode((x, y), h''_j)$
40: $ID_v \leftarrow$ ID of the node corresponds to point $(x, y)$ in the plane
41: $Action_v \leftarrow 0$ /* forward action */
42: Append $\langle ID_v, Action_v \rangle$ to $h''_j$
```

$ConstructSpanningTree$ on $G$ to generate a comb-like shortest-path spanning tree $t$, and then invoke the following three procedures: $SourceRoutePart1$, $SourceRoutePart2$, $SourceRoutePart3$ in lines 6, 7, and 8 respectively, on $t$, and $s_j$ as input arguments.

$SourceRoutePart1$ constructs the route to $v_1(x_1, y_1)$, which is the first-node of $s_j$, by traversing the comb-like shortest-path spanning tree $t$ using two **for** loops. In the first loop, line 20–22, we traverse the tree vertically by increasing the $y$ coordinate along the spine of the comb, i.e. $x = 1$ vertical line, starting from root-node $v_{1,1}$, until reaching the tooth of the comb with a $y$ coordinate equal to $(y_1 - 1)$. In the second **for** loop, line 23–25, we traverse the tree along the next tooth, i.e. the one with a $y$ coordinate equal to $y_1$, horizontally by increasing the $x$ coordinate till we hit $v_1$.

Each time through any of the two **for** loops, we append a forwarding node entry to $h_j$ by invoking another procedure, namely $AppendForwardingNode$ in line 21 and 24, which specifies the two elements of the appended node entry in $h_j$ as follows: ID of the node, and action value of 0, which denotes a forward action only. Then we append the forwarding node entry.

In $SourceRoutePart2$ (line 26–31), we iterate through the nodes of the anonymity-set $s_j$. In each iteration of the **foreah** loop, we specify the two element of the appended node entry as follows: the ID of the node, and the action value of 1, which denotes a respond action in a piggyback manner. Then we append the node to $h_j$.

In $SourceRoutePart3$, we construct the route from the last node of $s_j$ namely, $v_{|s_j|}(x_{|s_j|}, y_{|s_j|})$ to the root-node $v_{1,1}$. To achieve this, we traverse $t$ using two **for** loops. In the first loop, line 33–35, we traverse $t$ horizontally by decreasing the $x$ coordinate, starting from $x = x_{|s_j|} - 1$ till we reach the spine at $x = 1$. Then, in the second **for** loop on line 36–38, we traverse $t$ vertically by decreasing the $y$ coordinates along the spine, starting from $y_{|s_j|} + 1$ till we reach $v_{1,1}$. Similar to $SourceRoutePart1$, in each iteration of any of the **for** loops, we invoke $AppendForwardingNode$ to append a forwarding nodes entry to $h_j$ after specifying its two element: ID of the node, and action value of 0, which denotes a forward action only.

**Time Complexity**: The Source-Route Construction algorithm, listed in Algorithm 2, runs in time linear with the size of the square grid WSN, $|V| = n$. To see that, firstly we observe that $FindAnonymitySet$ takes at most $O(\lfloor n/k \rfloor) = O(n)$. The assignments and initialization, lines 2, 3, and 4, takes constant time, $\Theta(1)$. As we establish in section IV-A, the running time of $ConstructSpanningTree$ on a square grid graph is in $O(|V| + E||) = O(n)$. Each of the two **for** loops of $SourceRoutePart1$ takes $O(\sqrt{n})$. This is so because the first loop iterates along the spine of the comb-like spanning tree, and the second iterates along one of its teeth, and both are in $O(\sqrt{n})$ as mentioned in section IV-A.

The same argument is valid for the running time of $SourceRoutePart3$ with the parts of the comb-like tree along which the two **for** loops iterate are reversed. Therefore, $SourceRoutePart3$ takes $O(\sqrt{n})$. Lastly, $SourceRoutePart2$ runs in time linear with the size of the anonymity-set of the queried node $v_d$, i.e. it is in $\Theta(|s_j|)$. Since $|s_j| \in [k, 2k-1]$, and $k \in [1, n]$, thus $SourceRoutePart3$ takes time $O(n)$ in the worst case. Therefore, the total running time of the Source-Route Construction algorithm, listed in Algorithm 2, which is equal the sum of time taken by $FindAnonymitySet$, $ConstructSpanningTree$, $SourceRoutePart1$, $SourceRoutePart2$,and $SourceRoutePart3$ is $O(n + n + \sqrt{n} + n + \sqrt{n}) = O(n)$.

### C. The Querying Protocol

Our querying protocol, which implements the DAS anonymity scheme $(\pi, T, T^{-1})$, defines the rules for sending the client's query to a desired destination node $v_d$, and collecting its response anonymously. The protocol involves the following parties: the client $\mathcal{C}$, the sensor-cloud $\mathcal{S}$, and the WSN nodes $V$. $\mathcal{C}$ is the initiator of the protocol since she is the party who decides when to send a new query, the event that starts the execution of the protocol. Based on their action values in the header of the query packet, WSN nodes either forward the query packet to the next hop in the source-route header information, or piggyback their response data before forwarding.

In Protocol 1, we describe, in detail, how the parties engage in the querying protocol by exchanging messages, and acting upon receiving them. In the following, we identify some relevant aspects of the protocol that comply with the requirements of our trust model as specified in section II.

Recall from section II, in our trust-none model, the owner of the sensitive information namely, the client $\mathcal{C}$, trusts no other player in the anonymity scheme. In other words, the model allows the adversary to be outsider or any one of the querying protocol players except the owner of the sensitive information $\mathcal{C}$. Moreover, the scheme remains secure against the collusion of any of its outsider or insider adversaries. Consequently, in addition to the partition algorithm $\pi$, all of the anonymity transformations ($T$, and $T^{-1}$) are performed exclusively at the client $\mathcal{C}$. This is illustrated at the beginning of Protocol 1 where Algorithm 2, which utilizes the Partition algorithm $\pi$, is used to find the anonymity-set $s_j$ corresponds to the queried node $v_d$, and construct the source-rote header information $h_j$. In essence, this means that the anonymity transformation $T$ and the partition algorithm $\pi$ are run entirely at $\mathcal{C}$ before even contacting the sensor-cloud $\mathcal{S}$. Further, at the end of Protocol 1, the inverse anonymity transformation $T^{-1}$ is performed at $\mathcal{C}$.

## V. Performance Evaluation

The proposed DAS k-anonymous query scheme was examined via extensive simulation. We ensure all the targeted variables, such as average-case, and worst-case communication-cost, path length, and location-anonymity, are systematically measured with respect to the level-of-anonymity $k$, WSN network size $n$, and diameter $d$. We implemented all of our algorithms and querying protocol in a large-scale network environment on a dedicated 2.6 GHz Intel Core $i7$ x86_64 bit machine running OS X 10.11.6 El Capitan. In essence, we measured the following:

- How the anonymity scheme performance changes with the level-of-anonymity $k$ for various values of WSN network diameter $d$, and
- How the anonymity scheme performance changes with WSN network diameter $d$ for various values of level-of-anonymity $k$.

The query packet is structured as follows: 1 byte for the client ID, 10 bytes for MAC header, 1 byte for the action taken by each node to be either froward or append response then forward, 2 bytes for the node IDs. Each query data and response data is modeled to be constant at 32 bytes. To evaluate energy consumption, we adopt the MEMSIC TelosB mote platform [12] where each node is composed of a TI MPS

**Protocol 1** The Querying Protocol

```
upon receiving a request from the client $\mathcal{C}$ to query a destination node $v_d$:
 Using Algorithm 2:
   Find anonymity-set $s_j$ of $v_d$
   Generate corresponding source-route header information $h_j$
   Initialize $R$ to an empty list of size $|s_j|$
/*  $R$ stores piggybacked response data */
 Create a query packet $\rho$:
   Insert the client query data, and $R$ into the payload of $\rho$
   Insert $h_j$ into the header of $\rho$
 Send $\rho$ to the Sensor-Cloud $\mathcal{S}$
upon receiving $\rho$ by $\mathcal{S}$ from $\mathcal{C}$:
 Forwards $\rho$ to WSN through $v_{1,1}$ (the root-node of WSN)
upon receiving $\rho$ by an arbitrary WSN node $v$:
if $v$ is not the first node entry in $h_j$ then
    Drop $\rho$
else
    Remove $v$'s entry $\langle ID_v, Action_v \rangle$ from $h_j$
    if $Action_v = 0$ then
        $\rho \leftarrow \rho$ with $v$'s entry removed from $h_j$
        $Forward(\rho)$
    else if $Action_v = 1$ then
        Append $v$'s response data $d_i, i \in [1, n]$ to $R$ in $\rho$
        $\rho \leftarrow \rho$ with $v$'s entry removed from $h_j$
        $Forward(\rho)$
    end if
end if
upon receiving $\rho$ by $\mathcal{S}$ from $v_{1,1}$:
 Send back to $\mathcal{C}$
upon receiving $\rho$ by $\mathcal{C}$ from $\mathcal{S}$:
 Extract $v_d$'s response data from $R$
 Drop the rest of $\rho$
procedure $Forward(\rho')$
if $h_j$ is empty then
    Send back to $\mathcal{S}$
else
    Forward $\rho'$ to the next node in $h_j$
end if
```

430 microcontroller and CC2420 RF transceiver that consumes 1.8 microjoule per byte for reception and 2.1 microjoule per byte for transmission.

*1) Performance Metrics*

In addition to various performance anonymity trade-offs, we are interested in measuring the offered level-of-anonymity $k$ (the benefit) relative to the communication-cost $c$ (cost) of implementing the proposed scheme, as well as evaluating the achieved location-anonymity. For the benefit-cost analysis, the metric of Return-On-Investment (ROI) is introduced, where $ROI = \frac{k}{c}$. In essence, $ROI$ measures the return on investment relative to the cost of investment. $ROI$ is taken for both the average and worst-case communication-cost evaluation.

For the location-anonymity, two formal metrics are adopted to measure the average and maximum achieved anonymity. The first metric is the *radius of cloaking area* of a specific level-of-anonymity $k$ that is modelled as an equivalent circular area. Note that the cloaking area is the amount of space within which the true destination of a query is not identifiable due the the existence of its anonymity-set members, whereas the cloaking distance of a node, denoted as $d_c$, is the distance between it and any another node in its anonymity-set [28]. The radius $r$ measures the average cloaking distance of the centroid node over all anonymity-sets. Note that the centroid node of the anonymity-set $s_j$ is the node $v_i$ with the minimum sum of cloaking distances $Sum_i d_c$ to all other nodes in $s_j$, i.e., $\min_{v_i \in s_j}(Sum_i d_c)$. Thus, we compute the radius $r$ as the minimum value of the average of the cloaking distances of the centroid over all anonymity-sets for a specific $k$, i.e.,

$$r = \min_{s_j \in \mathbb{S}}\left(\frac{\min_{v_i \in s_j}(Sum_i d_c)}{k-1}\right), i \in [1, |s_j|], j \in [1, \lfloor n/k \rfloor]$$

To see how $r$ is computed, a simplified example in which $k = n = 9$, i.e. for a single anonymity-set case, is shown pictorially in Fig. 5.

Now, we show how to obtain the second metric, i.e. the maximum cloaking distance $d_{max}$. We first find the maximum cloaking distance $d_{max_i}$ for each node $v_i$ in an anonymity-set $s_j$ for a certain level-of-anonymity $k$. Since each node in the anonymity-set is a possible true destination, then we apply the concept of weakest link to select the minimum of the found maximum cloaking distances for each anonymity-set [29]. Eventually, we choose the weakest link again by taking the minimum over all of the anonymity-sets which results in the final maximum cloaking distance $d_{max}$. That is, $d_{max} = \min_{s_j \in \mathbb{S}}(\min_{v_i \in s_j}(d_{max_i})), i \in [1, |s_j|], j \in [1, \lfloor n/k \rfloor]$

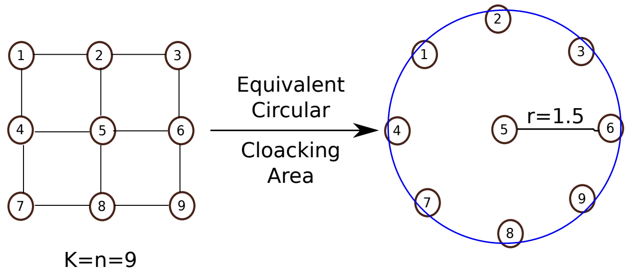


Fig. 5: A simplified example of creating the equivalent circular cloaking area.

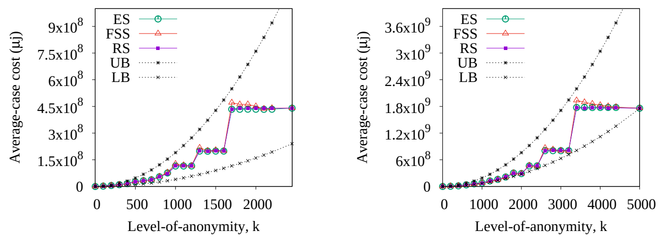
Based on the performance metrics, two sets of experiments are conducted with regard to the level-of-anonymity $k$ and the network diameter $d$.

## A. Level-of-Anonymity $k$

We first measured the average communication-cost $c_a$ incurred by DAS to achieve various level-of anonymity $k \in [1, n]$ based on a partition construction FSS, ES, and RS, respectively. Since the network diameter $d$ of a square grid graph is equal to $2(\sqrt{n} - 1)$, changing $n$ implies change of $d$. For more reliable results, our experiment is repeated in several WSN networks of sizes up to $n = 10000$, that is of diameter up to 198. In addition, we averaged the RS results over 100 random runs. We depict the obtained results for larger network sizes in Fig. 6.

It is observed that there exists a value of $k = d$, which we call the inflection point $k_0$, after which the average communication-cost is bounded from above and below by $O(k^2)$, and $\Omega(k^2)$ respectively, i.e. it is in $\Theta(k^2)$. Accordingly, these bounds are labeled as UB (Upper Bound) and LB (Lower Bound) in Fig. 6.

To explain the above results, we argue as follows. The communication-cost $c$ consists of two components namely, the routing-cost and the collection-cost, which is in $\Theta(d^2 + k^2)$ and $\Theta(k^2)$, respectively. Please see appendix for a formal proof (Lemma 1). Thus, the total incurred communication-cost $c$,



(a) n=4900, d=139 (b) n=10000, d=199

Fig. 6: The average communication-cost with varying $k$.

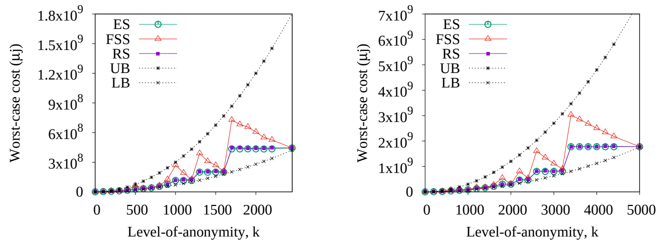
which is the sum of the routing-cost and collection-cost, is $c = \Theta(d^2 + k^2) + \Theta(k^2)$, which is simplified as follows.

$$c = \Theta(d^2 + k^2) \tag{1}$$

When $k$ is relatively large compared to $d$, the cost becomes $c = \Theta(k^2)$, which happens when $k$ exceeds the inflection point $k_0 = d$, as in the proceeding paragraphs.

In terms of comparing the different anonymity-sets construction methods, we observed that they are close to each other with respect to their incurred average communication-cost to achieve various values of $k$. It is expected that the FSS takes less energy because it tends to have the whole remainder in the first set close to the root node eliminating the first and third part of the source-route, as described in Algorithm 2. However, it was quiet surprising that FSS was outperformed consistently by the ES and RS at some points. These points occur when the remainder of $n/k$ is at its local maximum value. A reason for that must be related to the way that the piggy-back strategy collects sensor readings which incurs a cost that grows with the remainder of $n/k$. Additionally, the differences in energy consumption become less evident as the remainder approaches zero. This is what gives the curves their stair-like shapes.

Fig. 7 shows the results on worst-case communication-cost $c_w$. Similar observations are gained as that of the average-case above, where the worst-case cost of FSS was the highest, the Random-Spread performs in between the FSS and the ES, but closer to the later due to its random assignment of the remaining nodes to different sets.



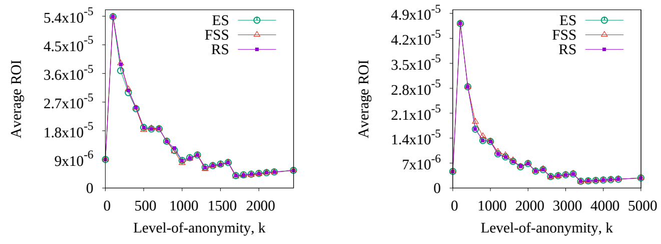
(a) n=4900 (b) n=10000

Fig. 7: The worst-case communication-cost with varying $k$ .

There is something specific and enriching that we discover by looking into the curves and data traces of $ROI$ for variable level-of-anonymity $k$ as shown in Fig. 8, and 9. The $ROI$ curves were shaped like a bell around the inflection point $k_0 = d$ for both the average and worst-case cost, and all the anonymity-sets construction methods. That is, the $ROI$ increased with $k$ till the inflection point $k_0 = d$ after which it decreased significantly with $k$. From practical viewpoint, this indicates that the ROI relative to the cost is at its highest values around the inflection point $k_0 = d$.

We argue as follows. First recall from equation 1 that the communication-cost $c$ consists of two components, one grows in proportion to $k^2$ and the other is in $d^2$. The first component increases with the anonymity-set size $|s|$ since $|s| \in [k, 2k-1]$. On the other hand, the second is due to the cost of forwarding a query to the first-node of anonymity-set, and returning collected responses from the last-node of the anonymity-set. Since we fixed the network size $n$ during this part of the experiment, the length of the forward and return paths shrinks with every increment in $k$. It follows that this component decreases with $k$. Now, for the range of $k < k_0$, The increment in the first component is marginal and balanced by the decrease in the second leaving the communication-cost almost constant. Thus, $ROI = k/c$ increases with $k$ for $k < k0$. Nevertheless, when $k$ exceeds the inflection point $k_0 = d$, the first component $k^2$ dominates the second which results in a substantial decrease in $ROI$ as the value of $k$ grows.

By comparing the $ROI$ data traces of different anonymity-sets construction method, we made comparable observations to that of the average and worst-case analysis mentioned in the proceeding paragraphs. That is, the values of the $ROI$ under the three methods were close to each other. However, the $ROI$ of FSS was the lowest in most of the time as $k$ increased due to the growing size of the first anonymity-set. This becomes more evident when the worst-case cost is used to compute the $ROI$ as shown in Fig. 9. Additionally, the $ROI$ of RS was between the FSS and the ES.

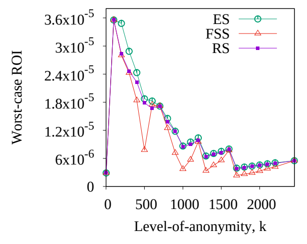


(a) n=4900 (b) n=10000

Fig. 8: The benefit-cost metric using the average communication-cost $ROI_{av}$ with varying $k$ .

Furthermore, we found that the total number of hops along the query and response path are in $O(k)$ for $k > d$ for both the average and worst-case, as shown in Fig. 10 and 11, regarding various $n$ and anonymity-sets construction methods.

The path length is further used as an index to compare the performance of the different anonymity-sets construction methods. We observed that all of them performed similarly in the average-case, which follows our intuition since the different anonymity-sets construction methods spread the same total number of hops in various ways. Thus, they produced comparable total number of traversed hops to query all the


(a) n=4900

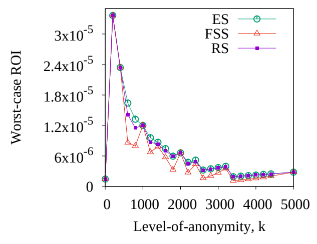
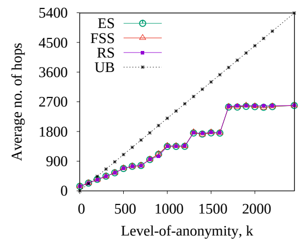

(b) n=10000

Fig. 9: The benefit-cost metric using the worst-case communication-cost $ROI_w$ with varying $k$

anonymity-sets and collect their responses. Nevertheless, the worst-case path length generated by the FSS is found longer than that by ES and RS whenever there is a remainder out of $n/k$. To see why, recall that the hops along the query and response path are of two types: routing-hops which increase with $d$, and collection-hops which increase with $k$. Now since the ES attempts to distribute the remaining nodes out of $n/k$ equally on all of the anonymity-sets, the worst-case collection hops grows steadily with $k$. On the contrary, the FSS adds all the remaining nodes to the first set. Thus, the worst-case number of hops occurs at the first set because it has the maximum number of collection hops. Since $d$ was fixed during this set of experiments, the collection hops dominated the routing hops when $k$ grows larger than $d$. This resulted in that the worst-case path length of FSS surpassed that of the ES whenever there is a remainder. On the other hand, RS acted closely to the ES because it disseminates the remaining nodes out of $n/k$ randomly on the anonymity-sets which makes it divergent from the FSS case.


(a) n=4900

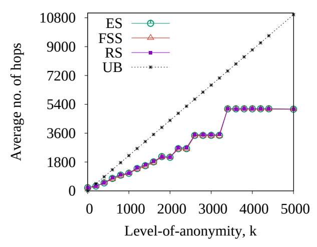
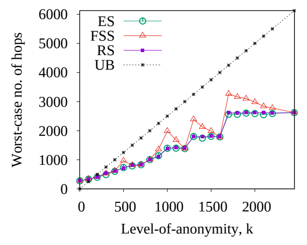

(b) n=10000

Fig. 10: The average number of hops with varying $k$ .


(a) n=4900

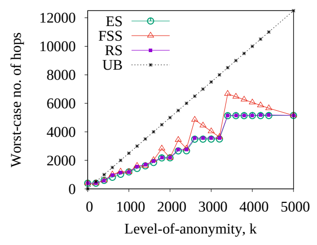
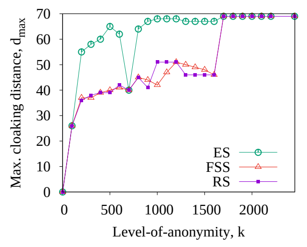

(b) n=10000

Fig. 11: The worst-case number of hops with varying $k$ .

We now evaluate the achieved location-anonymity by DAS anonymity scheme using the maximum cloaking distance $d_{max}$, and the radius of cloaking area $r$ metrics that are discussed in section V-1. As expected, we observed that both the $d_{max}$ and $r$ grew with $k$ because of the increase in the size of the anonymity-set. When $k$ is relatively large compared to $d$, the upper bound of $d_{max}$ and $r$ is in $O(d)$. Specifically, both $d_{max}$ and $r$ values are less than $d$. We depict our results for various network size $n$ in Fig. 12, and 13.

By analyzing the location-anonymity metrics data which is shown in Fig.s 12, and 13, we also noted that the ES achieved better location-anonymity compared to FSS whenever there is a remainder out of $n/k$. Nonetheless, RS stayed between them in most cases. To reason about it, recall that ES attempts to distribute the remainder equally on all the anonymity-sets while FSS adds all the remaining nodes out of $n/k$ to the first anonymity-set. Thus, the anonymity-sets created by FSS are all of the smallest size which is $k$ except for the first set. On the other hand, when the remainder of $n/k$ is greater or equal to $n/k$, ES yields anonymity-sets all of a size greater than $k$. Now, both of $d_{max}$ and $r$ adopts the security weakest link concept as discussed in section V-1 which always selects the anonymity-set with the minimum size to calculate these metrics. Hence, ES is in advantage by having the minimum size of the anonymity-sets greater than that of the FSS when there is a large enough remainder of $n/k$. In this way, the metrics of ES become higher than that of FSS as an indicative of better offered location-anonymity. Lastly, because of the randomness of RS, it acts between the two extremes.

As stated in section V-1, whereas $d_{max}$ measures the location-anonymity that is offered by the anonymity scheme, $r$ acts as a benefit-cost measure. In $r$, the benefit is sum of cloaking distances, and the cost is the level-of-anonymity which is proportional to the size of anonymity-set that provides the location-anonymity. In essence, $r$ is the radius of the equivalent cloaking area, and it gives a figure about how much location-anonymity is provided if a new node is added to the anonymity-set. By looking into the data traces of $r$, we discovered ES can provide a larger cloaking area than that by FSS but with less cost.


(a) n=4900

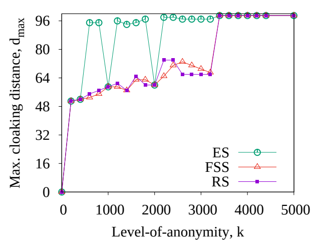

(b) n=10000

Fig. 12: The maximum cloaking distance metric $d_{max}$ with varying level-of-anonymity $k$ .

## B. Network Diameter $d$

The results of average and worst-case communication-cost obtained by varying $d$ for each value of $k$ are analyzed in this section. As shown in Fig. 14 and 15, when $d$ exceeds some point $d_0 = k$, the communication-cost in the average

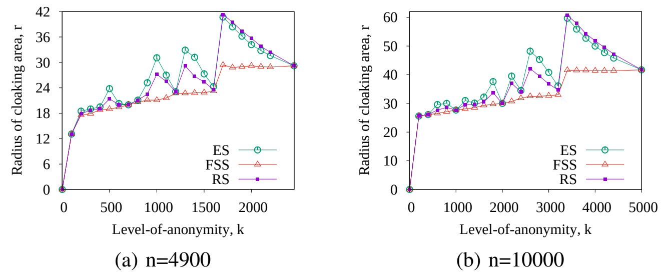


Fig. 13: The radius of equivalent cloaking area metric $r$ with varying level-of-anonymity $k$ .

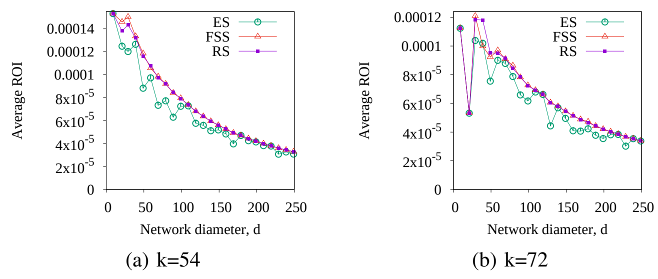


Fig. 16: The benefit-cost metric using the average communication-cost $ROI_{av}$ with varying $d$ .

and worst-case becomes bounded from above and below by $O(d^2)$ and $\Omega(d^2)$ respectively. This is fully coherent with the result of equation 1 which states that, $c = \Theta(d^2 + k^2)$. The upper and lower bounds are labeled on each graph as UB and LB respectively.

For the average case, it is observed that ES consumed more energy in the average-case as $d$ becomes larger compared to $k$, as shown in Fig. 14. Since FSS puts all the remainder of $n/k$ in the first anonymity-set, it is an energy saver with respect to $d^2$ component of the cost, and a consumer in terms of $k^2$ component of the cost. The opposite holds for ES. When $d$ surpasses $k$, the term $d^2$ becomes more dominant than $k^2$ component, and vice versa. Consequently, It is more practical to adopt FSS when $d$ exceeds $k$, as in figures 14, and 15. However, when $k$ is relatively large compared to $d$, ES should be preferred, as in 6, and Fig. 7.

For the worst-case results, as shown in Fig. 15, FSS was outperformed by ES specially at large values of $k$ due to the fact that the worst-case cost of FSS occurs in the first anonymity-set which vastly grows with $k$ when there is remainder.

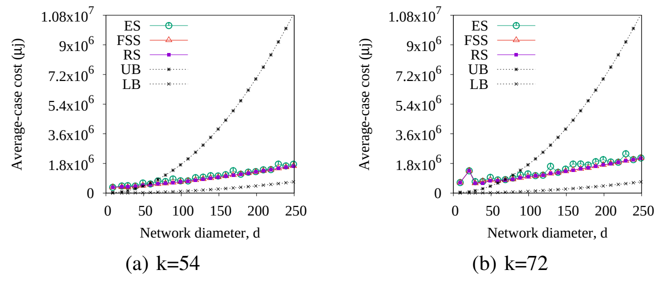


Fig. 14: The average communication-cost with varying $d$ .

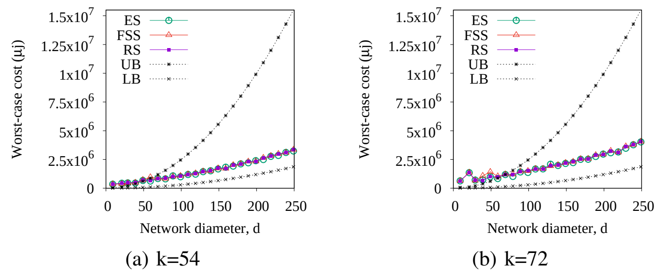


Fig. 15: The worst-case communication-cost with varying $d$ .

We have also found that $ROI$ decreased with $d$ for both the average and worst-case cost, as shown in Fig. 16, and 17, mostly due to the fact of $ROI = k/c$, where $c$ encompasses both the $k^2$ and $d^2$ components. Since $k$ is constant in this set of experiments, the sizes of anonymity-sets and the $k^2$ component of the cost varied slightly due to the mild change in the remainder of $n/k$. Therefore, approximately the $ROI$ changed inversely with $d^2$ when $d$ oversteps $k$. This became more obvious when $d$ grows beyond the inflection point $d_0 = k$ as $d^2$ dominates $k^2$.

In analogy with the average and worst-case cost results, the comparison among $ROI$ of different anonymity-sets construction methods of the anonymity scheme reveals similar implications. For the average-case results shown in Fig. 16, ES yields the lowest $ROI$ because of its sensitivity to the increment in the $d^2$ component of the cost. For the worst-case $ROI$ shown in Fig. 17, FSS achieves the lowest since the whole remainder was left in the first anonymity-set resulting in the highest worst-case cost, and hence the lowest $ROI$.

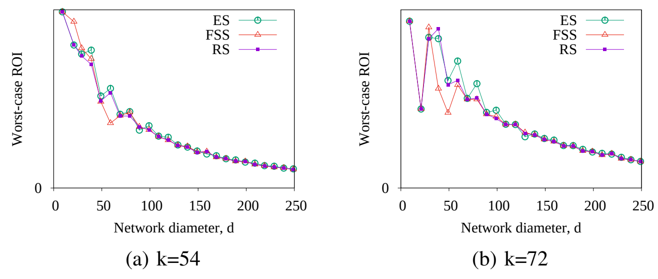
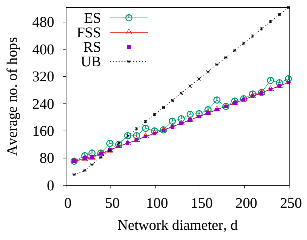


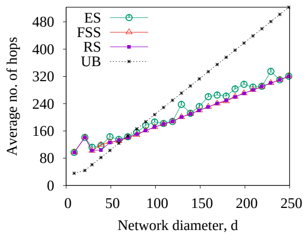
Fig. 17: The benefit-cost metric using the worst-case communication-cost $ROI_w$ with varying $d$ .

Fig. 18 and 19 show the path length results in the average- and worst-case scenarios for various values of $k$ under the three anonymity-sets construction methods. It is found that in the average-case, the generated paths by ES are mostly longer than that by the others, due to the fact that ES distributes the reminder nodes evenly to all the anonymity-sets, thereby subject to the longest path.

In the worst-case, on the other hand, the three anonymity-sets construction methods mostly produced identical path lengths as $d$ increased, mostly due to the fixed and relatively small values of $k$ that well mitigates the influence of collection-hops and lets the effect of routing-hops that grows with $d$ rather trivial. Hence, the worst-case path length occurred at an anonymity-set that is far away from the first set where the routing-hops are more. However, the size of such anonymity-set, and eventually the number of collection-hops, is almost the same for different anonymity-sets construction methods. This is because the remainder out of $n/k$ is small, so the anonymity-sets construction methods either change the size of the anonymity-sets other than the first set mildly or leave it untouched. Consequently, the different anonymity-sets construction methods behaved comparably.

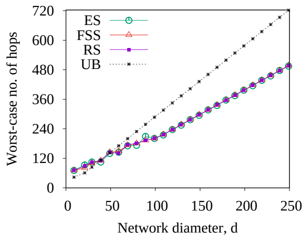


(a) k=54

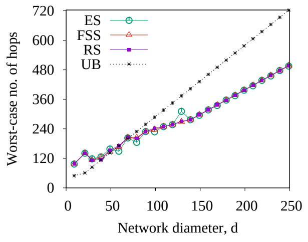


(b) k=72

Fig. 18: The average number of hops with varying $d$ .



(a) k=54



(b) k=72

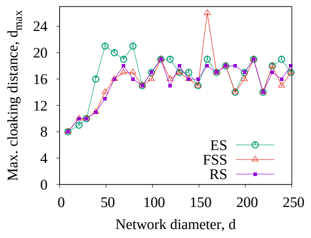
Fig. 19: The worst-case number of hops with varying $d$ .

Fig. 20, and 21 show the performance of location-anonymity of the proposed scheme. We observed that both of the location-anonymity metrics are smaller than $k$ when $d$ is relatively large compared to $k$. The reason is that the location-anonymity is limited by the anonymity-set size which is in turn asymptotically upper-bounded by the level-of-anonymity $k$.

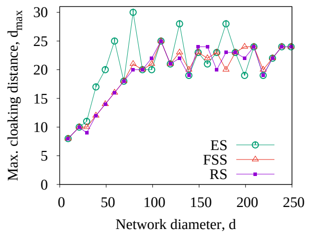
By comparing location-anonymity metrics under different anonymity-sets construction methods, we found that as $d$ is large they all performed similarly. This is due to the fact that as $d$ and $n$ are increased, the number of anonymity-sets $\lfloor n/k \rfloor$ exceeds the remainder out of $n/k$ since we kept $k$ fixed during this set of experiment. This possibly causes ES unable to distribute the remainder equally on all the anonymity-sets, and possibly some of the anonymity-sets would be left untouched at size of $|s| = k$ which is also the smallest size of anonymity-sets for FSS. Recall that we select the anonymity-set with the minimum size to compute $d_{max}$ and $r$. Thus, FSS and ES behaves similarly. However, they may differ a little due to the variations in the geometric structure of the anonymity-sets that affects the cloaking distances measurements within an anonymity-set.

## VI. RELATED WORK

Prior work in the area of analysis of trade-offs between query-anonymity and various performance metrics is limited

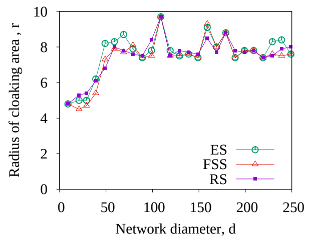


(a) k=54

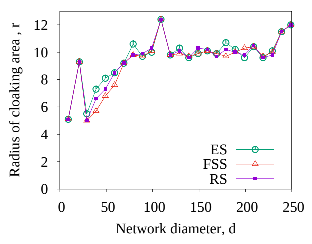


(b) k=72

Fig. 20: The maximum cloaking distance metric $d_{max}$ with varying network diameter $d$ .



(a) k=54



(b) k=72

Fig. 21: The radius of equivalent cloaking area metric $r$ with varying network diameter $d$ .

and not rigorous. Carbunar et al. [12], even in their simulation results, did not analyze the trade-off between query-anonymity and performance metrics along the whole range of level-of-anonymity. On the other hand, the work of De Cristofaro et al. [27] provides a querying protocol without characterizing the trade-off between its achievable query-anonymity and its incurred communication-cost. They measure the cost practically and independently from the obtained level-of-anonymity. Hayawi et al. [30] only focused on trade-off between query-anonymity and communication cost as a performance index for the query-anonymity scheme. As a step towards addressing this lack in the literature, we analyzed in-depth the trade-offs between the offered query-anonymity and various performance measures such as, communication-cost, path-length, return-on-investment metric, and achieved location-anonymity. Such comprehensive evaluation helps to unravel key factors affecting the performance of the anonymity scheme.

Chaum introduced the notion of anonymity-set in the DC-net [6] in 1988. The well-cited definition of anonymity-set is given by Pfitzmann and Hansen [7] , "*Anonymity* is the state of being not identifiable within a set of subjects, the anonymity-set." Considering anonymous communication systems which deal with sending and receiving of messages, anonymity is usually used to hide the identities of the actual sender, receiver or both within anonymity-set of other possible senders and/or receivers [7].

As stated in section III, the attacker intersects the output sequences of multiple runs of the querying protocol in order to identify the true destination of the client query [21], [31], [32]. In DAS, we adopt a unique approach that avoids intersection attack by using disjoint anonymity-sets with distinct members. This proactive approach has been stated briefly by De Cristofaro et al. [27] as follows,"we can let the client select the same route for the same queried node. Thus, ADV can not intersect its views of query executions", (ADV is the adversary). Nevertheless, they did not mention how to accomplish that. In this work, we maintain the disjoint property of the anonymity-sets by having the WSN nodes that connect the client to an anonymity-set forward the query but do not respond to it. To the best of our knowledge this is the first preventive measure to defend against intersection attack.

Although not in the context of trade-off analysis, schemes similar to DAS are studied in prior work [30], [33]. Therefore we do not claim novelty for the concept behind DAS. However, we presented DAS as a secure k-anonymous scheme, and studied its security properties by adopting information theoretic approach [34]–[37]. In addition, we break DAS down into its essential elements namely, the partition algorithm $\pi$, the anonymity transformation $T$, and the inverse transformation $T^{-1}$. This proves to be useful in our modular design of DAS for the sensor-cloud-based IoT environment, especially in the implementation of its partition algorithm, query routing algorithm, and querying protocol.

## VII. Conclusions

The paper studied the average- and worst-case query-anonymity performance trade-offs in the context of sensor-cloud-based IoT systems. We firstly presented the specification paradigm for the design and implementation of a secure k-anonymous query scheme for privacy preservation in the presence of unconditional eavesdropping adversary. Based on the proposed query and routing protocols, extensive analysis was conducted in order to establish bounds on trade-offs between offered query-anonymity and various performance measures such as communication-cost, return-on-investment $ROI$, path length, and offered location-anonymity. Our experimental results show that most of these bounds are functions of the level-of-anonymity $k$ and network diameter $d$, and the return-on-investment $ROI$ is at its highest values at the point $k = d$. In addition, the performance of the different anonymity-sets construction methods was observed and analyzed, which shows that in the average-case scenario, First-Set-Spread outperforms when $k$ is relatively small compared to $d$, while Equal-Spread is favored when $k$ is relatively large compared to $d$. In the worst-case scenario, First-Set-Spread fails behind Equal-Spread and this is more visible at large values of $k$.

We endeavor to make our evaluation results more comprehensive by testing our ideas in various simulation settings of heterogeneous anonymity-sets, mobile nodes, different routing algorithms, data collection strategies, network topologies and other system parameters. Another streak of research is to study all of the involved trade-off relationships such as communications-cost, cost-benefit metric, path length, return-on-investment, and location-anonymity metrics for the k-anonymous query scheme simultaneously.

## Appendix A

*Lemma 1:* Assume the query and response data are of constant size, to achieve a level-of-anonymity $k$ in a source-routed square grid whose diameter is $d$, the incurred communication-cost is $c = \Theta(d^2 + k^2)$.

*proof 2:* communication-cost $c$ consists of two components namely, the routing-cost and the collection-cost. The routing-cost is the cost of forwarding a query to the anonymity-set, and return collected responses from it. The collection-cost is the cost of collecting responses from all nodes in the anonymity-set. Assuming the query and response data is of constant size so that its transmission over one hope incurs a constant communication-cost, the collection-cost is in $\Theta(k^2)$, and the routing-cost is in $\Theta(d^2 + k^2)$. For data collection, we are using piggyback strategy in which each node of the anonymity-set $s_j$ appends its response to the query packet. Since the response data are of constant size by assumption, the collection-cost is incremented by $\Theta(1)$ over every next hop in the anonymity-set. Thus, the collection-cost is in $\Theta(1) + \cdots + \Theta(|s_j| - 1) + \Theta(|s_j|) = \Theta(|s_j|^2) = \Theta(k^2)$. The last equality comes from $|s_i| \in [k, 2k - 1]$, i.e. $|s_j| \in \Theta(k)$.

The routing-cost comprises two components namely, the return-cost of collected responses, and the source-route cost. To find the first, recall that the collected responses are of size $|s_j|$, and they travel a distance of $\Theta(d)$ back to the root-node along the third part of the rout (see Algorithm 2). This is so because the third part of the route is a shortest distance between a pair of nodes, and $d$ is the longest shortest distance between any pair by definition. This incurs a return-cost of $\Theta(d|s_j|) = \Theta(dk)$. For the source-route cost, recall that each node along the routing path removes its entry from the source-route header information $h_j$, as mentioned in our querying protocol listed in Protocol 1. Assuming the initial length of the $|h_j| = l$, we abstract this cost asymptotically to be equal to $\Theta(l) + \Theta(l - 1) + \cdots + \Theta(1) = \Theta(l) = \cdots + \Theta(l) = \Theta(l^2)$. To find $l$ in terms of $k$, and $d$, recall that $h_j$ consists of three parts from our routing algorithm listed in Algorithm 2. Each of the first and third is in $\Theta(d)$, since each of them is a shortest path between a pair of nodes, and $d$ is the longest shortest path between any pair by definition. The second path, which passes through the nodes of the corresponding anonymity-set $s_j$, is in $\Theta(k)$ because it is in $\Theta(|s_j|)$, and $|s_j| = \Theta(k)$. Thus, $|h_j| = l = \Theta(2d + k)$. Hence, the source-route cost is in $\Theta(l^2) = \Theta((2d + k)^2) = \Theta(4d^2 + 4dk + k^2) = \Theta(d^2 + k^2)$. Therefore, the routing-cost, the sum of responses return-cost and source-route cost, is in $\Theta(kd + d^2 + k^2) = \Theta(d^2 + k^2)$.

Now, the total incurred communication-cost $c$, which is the sum of the routing-cost and collection-cost, is $c = \Theta(d^2 + k^2) + \Theta(k^2)$, which is simplified as, $c = \Theta(d^2 + k^2)$.

## References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[3] X. Sheng, J. Tang, X. Xiao, and G. Xue, "Sensing as a service: Challenges, solutions and future directions," *IEEE Sensors journal*, vol. 13, no. 10, pp. 3733–3741, 2013.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[5] S. S. Mathew, Y. Atif, Q. Z. Sheng, and Z. Maamar, "The web of things-challenges and enabling technologies," in *Internet of things and inter-cooperative computational technologies for collective intelligence*. Springer, 2013, pp. 1–23.
[6] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *Journal of cryptology*, vol. 1, no. 1, pp. 65–75, 1988.
[7] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," 2010.
[8] R. Iyer, R. Rex, K. P. McPherson, D. Gandhi, A. Mahindra, A. Singh, and R. Raskar, "Spatial k-anonymity: A privacy-preserving method for covid-19 related geospatial technologies," *arXiv preprint arXiv:2101.02556*, 2021.
[9] D. Slijepčević, M. Henzl, L. D. Klausner, T. Dam, P. Kieseberg, and M. Zeppelzauer, "$k$-anonymity in practice: How generalisation and suppression affect machine learning classifiers," *arXiv preprint arXiv:2102.04763*, 2021.
[10] Y. Yan, E. A. Herman, A. Mahmood, T. Feng, and P. Xie, "A weighted k-member clustering algorithm for k-anonymization," *Computing*, pp. 1–23, 2021.
[11] K. Hayawi, "Cost analysis of query-anonymity on the internet of things," Ph.D. dissertation, University of Waterloo, Waterloo, ON, Canada, 2017.
[12] B. Carbunar, Y. Yu, W. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, pp. 14:1–14:34, Mar. 2010. [Online]. Available: http://doi.acm.org/10.1145/1689239.1689244
[13] S. W. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," 2020.
[14] B. Embrey, "The top three factors driving zero trust adoption," *Computer Fraud & Security*, vol. 2020, no. 9, pp. 13–15, 2020.
[15] J. Kindervag *et al.*, "Build security into your network's dna: The zero trust network architecture," *Forrester Research Inc*, pp. 1–26, 2010.
[16] M. Samaniego and R. Deters, "Zero-trust hierarchical management in iot," in *2018 IEEE international congress on Internet of Things (ICIOT)*. IEEE, 2018, pp. 88–95.
[17] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, no. 14, pp. 2826–2841, 2007.
[18] O. Goldreich, *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2004.
[19] K. Hayawi, P.-H. Ho, S. S. Mathew, and L. Peng, "Securing the internet of things: a worst-case analysis of trade-off between query-anonymity and communication-cost," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2017, pp. 939–946.
[20] O. Berthold, A. Pfitzmann, and R. Standtke, "The disadvantages of free mix routes and how to overcome them," in *Designing Privacy Enhancing Technologies*. Springer, 2001, pp. 30–45.
[21] D. Kedogan, D. Agrawal, and S. Penz, "Limits of anonymity in open environments," in *International Workshop on Information Hiding*. Springer, 2002, pp. 53–69.
[22] N. Mathewson and R. Dingledine, "Practical traffic analysis: Extending and resisting statistical disclosure," in *Privacy Enhancing Technologies*. Springer, 2005, pp. 17–34.
[23] E. F. Krause, *Taxicab geometry: An adventure in non-Euclidean geometry*. Courier Corporation, 2012.
[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.
[25] K. Bogart, S. Drysdale, and C. Stein, "Discrete math for computer science students," 2004.
[26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.
[27] E. De Cristofaro, X. Ding, and G. Tsudik, "Privacy-preserving querying in sensor networks," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, Aug 2009, pp. 1–6.
[28] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, 2003, pp. 31–42.
[29] C. P. Pfleeger and S. L. Pfleeger, *Security in computing*. Prentice Hall Professional Technical Reference, 2002.
[30] K. Hayawi, A. Mortezaei, and M. V. Tripunitara, "The limits of the trade-off between query-anonymity and communication-cost in wireless sensor networks," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '15. New York, NY, USA: ACM, 2015, pp. 337–348. [Online]. Available: http://doi.acm.org/10.1145/2699026.2699113
[31] J.-F. Raymond, "Traffic analysis: Protocols, attacks, design issues, and open problems," in *Designing Privacy Enhancing Technologies*. Springer, 2001, pp. 10–29.
[32] D. Agrawal and D. Kesdogan, "Measuring anonymity: The disclosure attack," *IEEE Security & privacy*, vol. 1, no. 6, pp. 27–34, 2003.
[33] L. von Ahn, A. Bortz, and N. J. Hopper, "K-anonymous message transmission," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, ser. CCS '03. New York, NY, USA: ACM, 2003, pp. 122–130. [Online]. Available: http://doi.acm.org/10.1145/948109.948128
[34] C. E. Shannon, "Communication theory of secrecy systems*," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.
[35] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, ser. PET'02. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 54–68. [Online]. Available: http://dl.acm.org/citation.cfm?id=1765299.1765304
[36] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, ser. PET'02. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 41–53. [Online]. Available: http://dl.acm.org/citation.cfm?id=1765299.1765303
[37] G. Tóth, Z. Hornák, and F. Vajda, "Measuring anonymity revisited," in *Proceedings of the Ninth Nordic Workshop on Secure IT Systems*. Espoo, Finland, 2004, pp. 85–90.

**Kadhim Hayawi** Kadhim Hayawi is a Graduate Program Coordinator and an assistant professor at the College of Technological Innovation at Zayed University, and a member of the Cybersecurity and Digital Forensics research group where he teaches a wide variety of courses and pursues his research endeavors. He received his M.Sc. degree in Computer Science from Dalhousie University, Canada in 2004, and a Ph.D. degree from University of Waterloo, Canada in 2018. He earned several prestigious industry certifications and has over 20 years of experience in academia, and industry. His research interests are in tackling Information Security and Privacy challenges of Emerging Technologies such as IoT, Industrial IoT, Fog, Cloud, and Social Networks using Real-Time and Distributed Deep Learning, GAN, and Blockchain Technologies.

**Pin-Han Ho** Dr. Pin-Han Ho is currently a full professor in the Department of Electrical and Computer Engineering, University of Waterloo. He is the author/co-author of over 400 refereed technical papers, several book chapters, and the coauthor of two books on Internet and optical network survivability. His current research interests cover a wide range of topics in broadband wired and wireless communication networks, including wireless transmission techniques, mobile system design and optimization, and network dimensioning and resource allocation. He is in the rank of IEEE Fellow and a Professional Engineer Ontario (PEO).

**Sujith Samuel Mathew** DR. Sujith Samuel Mathew completed his Ph.D. in Computer Science from the University of Adelaide, South Australia. He has twenty years of experience working both in the IT Industry and in IT Academia. He has held positions as Group Leader, Technical Evangelist, and Software Engineer within the IT industry. In academia, he has been teaching various IT related topics and pursuing his research interests in parallel. His re- search interests are in ubiquitous and distributed computing, with focus on the Internet of Things and Smart City applications. Presently, he is an Assistant Professor at the College of Technological Innovations (CTI), Zayed University in Abu Dhabi, UAE.

**Limei Peng** Limei Peng is an Associate Professor with the School of Computer Science and Engineering, Kyungpook National University (KNU), Daegu, South Korea. Before she joined KNU, she was an Assistant Professor with the Department of Industrial Engineering, Ajou University, Suwon, South Korea, from 2014 to 2018, an Associate Professor with Soochow University, China, from 2011 to 2013, and a Postdoctor Fellow of the Grid Middleware Research Center, Korea Advanced Institute of Science and Technology (KAIST), South Korea, from 2010 to 2011. Her research interests include cloud computing, fog computing, datacenter networks, the IoT, and 5G communication networks.