Privacy Preservation Schemes for Querying Wireless Sensor Networks

Tassos Dimitriou
Athens Information Technology
19.5 km Markopoulo Ave., 19002, Peania
Athens, Greece
Email: tdim@ait.edu.gr

Ahmad Sabouri
Chair of Mobile Business & Multilateral Security
Goethe University Frankfurt
Frankfurt, Germany
Email: ahmad.sabouri@m-chair.net

***Abstract*—In this work we study the problem of query privacy in large scale sensor networks. Motivated by a novel trust model in which clients query networks owned by trusted entities but operated by potentially untrusted adversaries, we consider several proposals for protecting the identities of the queried sensors.**

Our schemes do not rely on the use of public key cryptography nor do they make any assumptions on the topology of the network. Inspired by the data-centric communication model and the collaborative nature of sensor networks, our proposals distribute the data in random locations to be later retrieved by random direction queries, using only local computations and total absence of coordination. It is perhaps of interest to note that query privacy is achieved using only lightweight, symmetric cryptographic primitives.

Extensive analytical and experimental results confirm that the proposed protocols can achieve their goals using only minimal communication and storage overhead.

I. Introduction

Wireless Sensor Networks (WSNs) have been an attractive paradigm for pervasive computing oriented applications. WSNs have found wide applications in areas as diverse as environmental and habitat monitoring, healthcare, home automation, and traffic control.

To accomplish the targeted functionalities, large scale WSNs usually collect or generate vast amounts of data during their lifetime. However, the traditional trust model where the owner and consumers of the network are considered to be the same does not make sense for such large scale networks. For example, NEPTUNE [1], GEOSS[2], and ORION[3] are building ocean observatories and systems for observing and reporting earth, ocean and atmosphere information. The trust model of these networks is shaped by two factors. First, multiple organizations are involved, acting both as funding sources and primary investigators. Even though network owners may need to collaborate for administrative purposes, they don't have to fully trust each other due to diverging interests. Second, external organizations interested in the areas monitored by the sensor network may be willing to pay for access to sensor readings [13].

The type of networks described above introduces new concerns regarding privacy for the end-users. Since end-users are accessing the network via services provided by the network owners and operators, information about their identity along with their access pattern or the region of interest can be abused by the servers against the user's interest.

In this work we consider the problem of query privacy in outsourced WSNs. Motivated by robustness, scaling and energy efficiency aspects, we propose a number of schemes that draw their inspiration from the *Data-Centric* (DC) communication paradigm of sensor networks [4]. Our schemes are based on the observation that if sensor readings are distributed in random locations in the network and queries are sent towards random destinations, query privacy can be enforced even under the presence of malicious nodes in the network. We compare our findings against previous work and we validate the performance and the effectiveness of our algorithms using both analytical and experimental results.

The rest of this paper is organized as follows. Section II discusses previous methods along with their strong and weak points. The network model, assumptions and the privacy problem we consider in this work are stated in Section III. Section IV, gives the details of our schemes and an analytical model for the costs incurred. A data tagging scheme is necessary to achieve the desired privacy properties; this is discussed in Section V. Several issues related to the proposed protocols are discussed and validated experimentally in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Providing anonymity across untrusted networks has been studied in various contexts ([5], [6], [7]). Onion Routing[8] principles also became the basis of many schemes such as TOR[9] for providing anonymity in untrusted networks.

Regarding WSNs, an initial attempt to provide anonymity of network structure has been made in [6] where the network protocol leverages a dynamic virtual infrastructure on top of the physical layer. Anonymity in clustered wireless sensor networks has been studied in [10] and two anonymous schemes have been proposed. However, these schemes assume the secret keys shared between sensor nodes and the base station are not compromised and that nodes inside a cluster are not distinguishable. Assuming knowledge of the location by the nodes and the ability of direct communication with the base station, [5] provides two techniques of anonymity based on one-way keyed hash chains. It is possible, however, for an adversary to eavesdrop on the communication and compromise a node's data and encryption keys [11].

Another work on anonymous data collection [12] is based on perturbations but does not use provably secure cryptographic techniques. In order to provide query privacy in WSNs, [13] proposed a scheme using two servers. This approach, however, is not efficient in terms of storage because each node must share a key with a client. Furthermore, privacy is completely lost against colluding servers.

A recent work on privacy-preserving querying in WSNs was presented in [11] based on the Onion Routing methodology. This scheme, however, is based on asymmetric cryptography which is not a good option for low processing power devices. Furthermore, the onion routing scheme results in a large overhead for the message size and leads to high power consumption since transmission is the most costly operation in sensor nodes. Finally, knowledge of the network topology is assumed which is not the case in our proposed schemes.

III. Network Model and Assumptions

The network setting we consider here consists of N nodes denoted by $\{s_1, s_2, ..., s_N\}$. Sensors are considered to have some storage facilities to save their own measured data or the data received from others.

There are three distinct entities participating in the protocol namely the network owner, the network operator and the client which we refer to them as $\mathcal{OWN}$, $\mathcal{OPR}$ and $\mathcal{CLN}$, respectively. $\mathcal{OWN}$ is the entity who initializes the nodes and may share some keys with them. $\mathcal{OPR}$ provides the gateway, $\mathcal{GW}$, to the clients in order to access the network. Any query which goes through the network must be submitted to $\mathcal{GW}$. Responses to queries are returned to $\mathcal{GW}$ for delivery to the client. The privacy problem in this work follows the same formulation as in [11]. Since $\mathcal{OPR}$ owns $\mathcal{GW}$, it will be considered as a potential adversary. The challenge here is to ensure that the identity of the queried sensor cannot be disclosed by the adversary.

In this paper we consider an adversary $\mathcal{ADV}$ as an *honest-but-curious* one who controls both $\mathcal{OPR}$ and $\mathcal{GW}$ [11]. It can eavesdrop on the packets but does not actively interfere with the specified behavior of the protocol. Furthermore $\mathcal{ADV}$ is considered to be able to compromise z nodes in the network, where $0 \ll z \ll N$. Once a node is infected, $\mathcal{ADV}$ learns all of its secret keys and can thus obtain the plaintext of all incoming and outgoing messages. We note that the privacy of $\mathcal{CLN}$ is out of scope for this work. $\mathcal{CLN}$ does not care if $\mathcal{ADV}$ knows that she is querying the network.

IV. Privacy Preservation Schemes

External Storage model: The simplest model that could be used to achieve query privacy is to have an external storage server which belongs to the trusted party $\mathcal{OWN}$ and receives data collected by the sensors. In this case all queries are submitted to the storage server. Here the network follows the *report-on-sense* model; all nodes send their data constantly at a cost of $O(\sqrt{N})$ per event. Since there is no need to keep any information internally, the cost for local data storage is $O(1)$. Overall, this scheme is impractical if the sensing rate of sensors is large compared to number of queries. Privacy here reduces to the Private Information Retrieval (PIR) [14] problem which has been well studied in the literature but has not not reached the point of being practical yet [11].

Local Storage model: In this model, event information is stored locally at each node. However, queries must be flooded to all nodes at a cost of $O(N)$. To avoid privacy loss, the client must query and receive data from the *entire* set of sensors since otherwise it would be easy to spot the reporting node. This would achieve perfect privacy but would result in a massive waste of energy. To avoid this problem, the authors in [11] assume knowledge of the network topology and use the onion routing technique to target a specific node. The goal is to reduce communication costs while at the same time achieving a good level of privacy.

Data-Centric model: In the data-centric (DC) model, data are "named" by attributes; a node storing information about an event is determined by the event's name. Thus, all data related to similar events will be stored at the *same* node (not necessarily the node that originally gathered the data) [4]. The advantage of this approach is that queries for data can be sent directly to the node storing these named data, at a cost of $O(\sqrt{N})$, thereby avoiding the query flooding typically required in other proposals. Unfortunately, this method is unattractive from a privacy point of view. Addressing a specific storage node can leak information about the original one as the mapping between sensing nodes and storage ones is deterministic and well known in advance. In what follows, we will address this problem while at the same time maintain the communication benefits of the DC approach.

A. Random Direction Query

The idea is to abandon the deterministic model of addressing nodes and utilize the routing topology of the network to select "witnesses" for a node's data. In particular, we will propose protocols that randomize the witnesses for a given node's readings (by *replicating* these readings), so that the adversary cannot anticipate their identities. An implicit benefit of this approach is that we no longer need to rely on knowledge of the network topology as in [11].

For this approach to work there must exist a *tagging mechanism* so that witnesses can identify requested data *without* disclosing any information about the originator. A description of such a tagging mechanism is deferred to Section V. Once such a mechanism is in place, random queries can be used to obtain the required data without revealing the identity of the queried node. We can organize the details of this approach in the following three steps:

Sensing: Upon a new reading, each node selects α *random* locations and sends its data to the closest nodes to each location using a geographical routing algorithm such as GPSR [15]. Data is encrypted using a key shared between the sensor and $\mathcal{OWN}$ before deployment, and tagged appropriately (Section V). Encrypted data does not disclose the identity of the originator so even the direct neighbors of the node cannot realize if it is fresh data or forwarded ones.

The tag transmitted with each encrypted reading gives the opportunity to match the query with the data. The originator stamps the data with an expiration time so that replicas can be discarded after expiration.

Querying: There is a server $\mathcal{S}$ owned by the trusted party where $\mathcal{CLN}$ can request for a query to be properly generated. Therefore $\mathcal{CLN}$ can easily take this query and submit it to $\mathcal{GW}$. $\mathcal{GW}$ will forward this query to β *random* destinations hoping to hit one of the replicas.

Replying: When a node receives a query, it analyzes the tagging information included in the query to check if it has the requested data available in its storage. In the case of matching tag, the node replies back with the encrypted data.

In order to have a successful query at least one of the β selected destinations must be among the α replicas. The query will fail when all the β selected nodes are from the $N-\alpha$ non-replicas. Thus, the probability of failure is $C(N-\alpha,\beta)/C(N,\beta)$, which can be upper bounded by $(1-\alpha/N)^{\beta}$ or $e^{-\frac{\alpha\beta}{N}}$, given that $(1-x) < e^{-x}$, for $x < 1$. Hence the probability P_s of having at least one of the queries meeting the data is $P_s \geq 1 - e^{-\frac{\alpha\beta}{N}}$. For example, when $\alpha\beta = N$ or $\alpha = \beta = \sqrt{N}$, P_s is at least 63% and grows rapidly for larger values of α, β.

Regarding resources, on average each replica node must support $O(\alpha)$ readings. Thus, in a 10000 node network, a random node will be the recipient of about a hundred readings from nodes that sense and transmit their data. Also, due to each replication process, there are α point-to-point communications per sender which leads to a cost of $O(\alpha\sqrt{N})$ for sensing plus $O(\beta\sqrt{N})$ for querying, or $O(N)$ if $\alpha = \beta = O(\sqrt{N})$.

Figure 1 depicts how this scheme works when $\alpha = 2$ and $\beta = 3$. The originator node indicated by a solid black circle has replicated its data in $R1$ and $R2$. $\mathcal{GW}$ has sent the query to the three destinations $D1$, $D2$ and $D3$. As it is shown, only the third query has met the required data.

B. Random Direction Query^{+}

To reduce the costs of the previous protocol, we investigate a different scheme inspired by the work on Rumor Routing [16]. We note that nodes in a sensor network operate both as sensing devices and as routers. For a sensor's reading (or query) to travel from note u to node v, it must pass through a number of intermediate nodes. If these intermediate nodes also store the sensor data, effectively a "line" is drawn across the network. If a query ever crosses such a line, then the data will reach the client much faster than in the first case. This will result in significant savings in communication costs as we explain below.

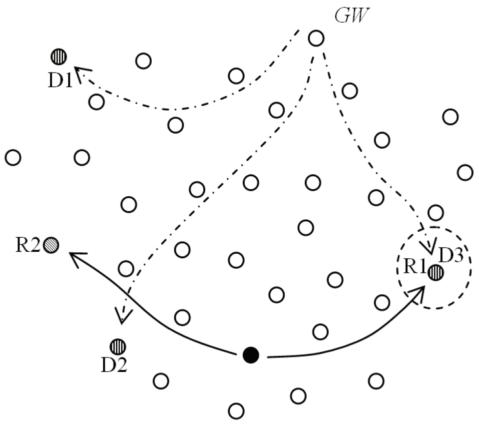

Figure 1. Random Direction Query: $\alpha = 2$, $\beta = 3$

It can be shown ([17]) that the expected number of intersections of r randomly drawn lines intersecting within the bounds of the unit circle is given by $r(r-1)\left(\frac{1}{6} + \frac{245}{144\pi^2}\right)$. So, with three lines, we expect to have two collisions. Similarly[16], Monte-Carlo simulations show the probability of two lines intersecting in a bounded rectangular region to be approximately equal to 69%. So, in our enhanced query scheme, we are going to achieve a high probability of hitting the required data by replicating them on *all* the nodes in the path toward the α' selected locations as well as checking *all* the intermediate nodes in the β' query paths. 69% probability of success is for the case where $\alpha' = \beta' = 1$.

The transmission cost now is reduced by factor of $O(\sqrt{N})$ for sensing and querying. Since we need to keep the tagged data on all the nodes in the path, the storage cost per replica still remains equal to $O(\sqrt{N})$, on average.

The same example is used as before (Figure 1) but this time, all the nodes in the path towards $R1$ and $R2$ keep a copy of data in their storage and all the nodes in the query path will be investigated for the required data. In this case, the query toward $D2$ also meets the desired data in its way.

C. Approximate Costs & Comparison

In this section we develop an analytical model to compare the various schemes. Let D denote the total number of readings (sensed events) and Q the total number of queries. Table I provides the overall communication costs of our schemes in terms of D, Q, α, β, α', β' and N. Although the external server case reduces to the PIR problem, it is included here for comparison.

Table I
COMMUNICATION COSTS

	Transmission
External Storage Server	$D\sqrt{N}$
Random Direction Query $\alpha = \beta = \sqrt{N}$	$\alpha D\sqrt{N} + \beta Q\sqrt{N} = O((D+Q)N)$
Random Direction Query+ $\alpha' = \beta' = 1$	$\alpha' D\sqrt{N} + \beta' Q\sqrt{N} = O((D+Q)\sqrt{N})$

For example, in the second row we see that for every sensed event, the sensing node must replicate the event in α other sensors for a total cost of $\alpha D\sqrt{N}$. Similarly, every query must be disseminated to β random locations for a total cost of $\beta Q\sqrt{N}$. The same is true in the third row but now both α', β' are constant.

To provide a comparison with the work in [11], however, we need to take into account the *packet size*. The work in [11] follows the Onion Routing technique in which case the packet carries information for *all* the nodes in path, therefore the one-hop transmission cost is not the same as in our schemes (in our case a packet carries just one sensor's data). In particular, the authors required the size of onion to be always the same in order not to disclose any information about the distance to the target. Thus the onion needs to have a size of $O(\sqrt{N})$. In what follows we define a new metric, the Effective Transmission Load (ETL), as

$$ETL = C_T \times P,$$

where C_T is the transmission cost of a *constant* size packet and P the packet size. For the schemes presented in this paper the packet size is $O(1)$. As a result we can summarize the ETL factor for each of the schemes in Table II.

Table II
EFFECTIVE TRANSMISSION LOAD (ETL)

	ETL
External Storage Server	$D\sqrt{N}$
Random Direction Query	$(\alpha D + \beta Q)\sqrt{N}$
Random Direction Query+	$(\alpha' D + \beta' Q)\sqrt{N}$
Onion Routing approach [11]	$Q2\sqrt{N}\sqrt{N} = 2QN$

The simplest Random Direction Query scheme compares favorably with the scheme in [11], provided the total number of readings D is smaller than the number of Q of queries in the network. This can be the case of a frequently queried sensor network.[1] The enhanced scheme is a clear winner in terms of communications costs (case $\alpha' = \beta' = 1$) by almost a factor of $O(\sqrt{N})$. The larger storage costs in these cases maybe be considered a necessary tradeoff to account for the heavier asymmetric cryptography computations and network topology knowledge required in [11].

V. TAGGING SCHEME

In this section we consider the details of our tagging scheme. Let $E_i(d)$ and $T_i(d)$ denote the encrypted data and tag sent by sensor s_i to a set of random replicas. When $\mathcal{CLN}$ wishes to query s_i for data, it makes a request to the trusted server S operated by $\mathcal{OWN}$ and a tag T^* is generated which is forwarded to $\mathcal{GW}$. T^* is then disseminated to a set of random destinations according to the methods described in the previous section. If a replica is found such that the stored $T_i(d)$ equals T^*, the corresponding encrypted data $E_{K_i}(d)$ is transmitted back to $\mathcal{GW}$ and eventually to $\mathcal{OWN}$ which upon decryption delivers it to the client.

Obviously, if the tagging scheme is deterministic there is a basic, *brute force* attack: an adversary records T^* and then queries S for sensor data until the server returns the target value T^*. In that case, $\mathcal{ADV}$ learns the ID of the queried node. Thus, we need to ensure that the tagging scheme is *probabilistic* so that even if the same sensor is queried twice, the generated tag delivered by S to $\mathcal{GW}$ will be *different*.

The concept of indistinguishability is well studied in the literature and comes under the name of semantic security [18]. Intuitively, if an encryption system possesses the property of indistinguishability, an adversary will be unable to distinguish pairs of ciphertexts based on the message they encrypt. Thus, it makes sense to use a semantically secure encryption scheme to generate the tags of the messages (here we don't have to use a Message Authentication Code scheme since, according to our threat model, the service we want to provide is *neither* integrity *nor* authenticity of transmitted messages). The details of our scheme follow.

Let s_i be the sensor under question and let K_i be the key it shares with the trusted server (preloaded before deployment). For simplicity we assume that sensing occurs in phases, so during phase j, s_i senses data d_j. Then s_i first produces a ciphertext $c = E_{K_i}(\text{``}data\text{''}, d_j, i, j)$ of the data and then a tag $\tau = E_{K_i^j}(\text{``}tag\text{''}, j)$. Both c and τ are disseminated to a set of random replicas.

The key K_i^j used in the construction of the tag is a *phase-dependent* key produced by the operation $K_i^j = F(K_i, j)$, where $F()$ is a secure one-way hash function. Upon query by a client, the trusted server can generate this key on the fly using the key it shares with s_i and knowledge of the phase j. Then it can also generate $\tau^* = E_{K_i^j}(\text{``}tag\text{''}, j)$ that will be forwarded to $\mathcal{GW}$ for propagation to a set of random destinations. Notice that indistinguishability of ciphertexts guarantees that no information leaks about the identity of the queried sensor. Thus the basic attack cannot longer be applied. The question, however, becomes: how can a replica node r match the tag propagated by $\mathcal{GW}$ with the one stored in its memory if tags are different?

If key K_i^j is known to the replica r, then r can perform a trial decryption of all the tags in its memory and see if it gets back a valid plaintext ("tag"). If this is the case, it

[1] Independence of D in the Onion routing case is attributed to the use of the *read-on-demand* model. In this model a node never senses continuously but is tasked to sense only when instructed. Thus a comparison with our schemes is meaningful only with respect to the cost of *querying* the network for data.

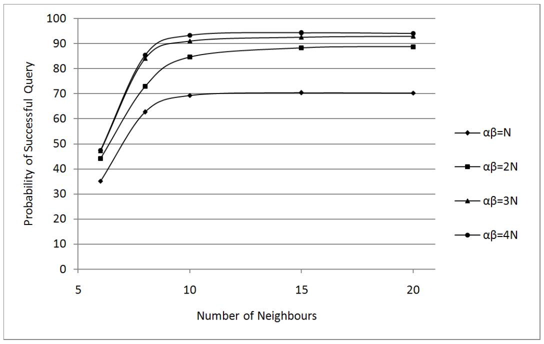

Figure 2. Simulation Results for Random Direction Query Scheme

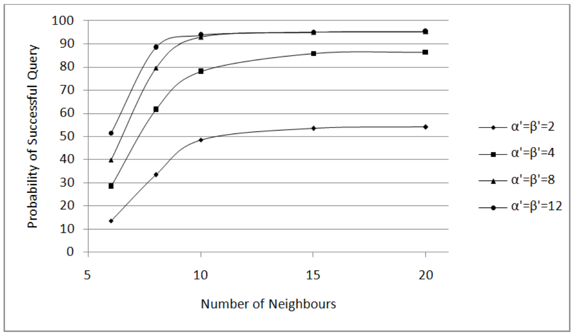

Figure 3. Simulation Results for Random Direction Query$^+$ Scheme

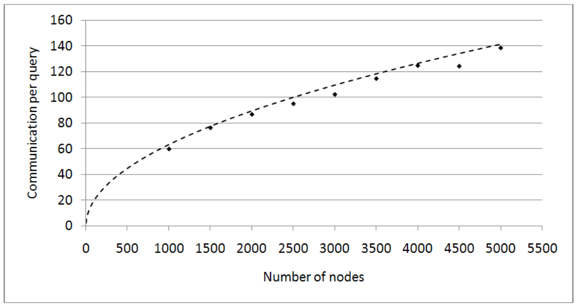

Figure 4. Growth of Communication Cost per query for RDQ$^+$ when $\alpha' = \beta' = 4$ and density factor is 15

has found a matching tag and can respond back with the ciphertext $E_{K_i}(\text{“}data\text{”}, d_j, i, j)$. So, it is necessary for r to get hold of the one-time key K_i^j. Although in principle it is possible to establish keys on the fly between nodes in a sensor network ([19], [20]), we will refrain from doing so. Instead, we will allow s_i to transmit K_i^j in the *clear* once the phase is over. Sensor s_i will simply send the tagging key to the same sensors as it sent its encrypted data and tag.

This is safe to do since the tag does not contain any data that can relate a tag with a node's ID. A malicious node upon decrypting such a tag will only get the phase value under consideration. But this is the same for all nodes sensing data during phase j. This is also a direct consequence of our threat model since the adversary neither modifies or blocks messages nor it interferes with a sensor's behavior.[2] So, releasing the tagging key does not have any impact of the privacy of the queried nodes.

VI. Experimental Results

In this section we evaluate the effectiveness of the proposed schemes. We generated random network topologies by placing 1000 nodes uniformly at random in the unit square with a network density (i.e. number of neighbors) ranging between 5 and 20. To ensure statistical validity, each experiment was repeated 100 times in which every node replicated its data towards α random locations and queries for randomly selected nodes were sent towards β random directions.

We have simulated the Random Direction Query (RDQ) scheme having $\alpha \times \beta = N, 2N, 3N$ and $4N$. Figure 2 depicts the query success probability as a function of the network density. Clearly the values of α and β must be in the order of $\sqrt{N}$ in order to achieve a considerable query success rate. The other important factor is network density and connectivity. As the number of neighbors increase, we will have a better chance to route a replication and a query to the same node. The results of these experiments imply that in order to have a success probability of 90% we should have at least $\alpha \times \beta = 3N$ and density factor around 10.

Evaluation of the Random Direction Query$^+$ (RDQ$^+$) scheme is demonstrated in Figure 3 for $\alpha' = \beta' = 2, 4, 8$ and 12 and varying network densities. When compared with the previous scheme we can see that the case $\alpha' = \beta' = 8$ is as effective as the case $\alpha \times \beta = 4N$ when the network is dense enough. The savings here are a $\sqrt{N}$ factor in communication overhead. Figure 4 illustrates that the communication cost per query grows as a function of $\sqrt{N}$, where $\alpha' = \beta' = 4$ and the density factor is 15. We have also measured the average storage requirements per node in this experiment and the results (Figure 5) verify that the amount of required storage is again proportional to $\sqrt{N}$.

A. Further Enhancements

The RDQ$^+$ scheme was further analyzed considering the fact that whenever a node transmits some information to one of its neighbors, the rest can hear this message. So, we modified the RDQ$^+$ scheme in such a way to take advantage of this overhearing when forwarding a query to the next hop. As a result, the probability of hitting a replica by a query increases by 5-10%, depending on the case, without any additional storage or communication overhead (details omitted due to space restrictions).

[2] The only case that a malicious node u can tell that a particular tagging key originates from a specific node v is if it overhears transmissions and discovers that v was never the receiver of such a key. But this is highly unlikely in a sensor network where nodes have comparable radio ranges since it would require that u and v have exactly the same neighbors.

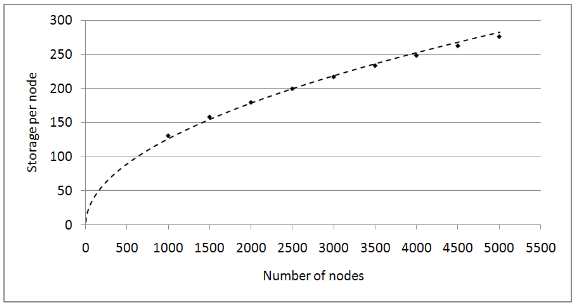

Figure 5. Growth of Storage per reading for RDQ$^+$ when $\alpha' = \beta' = 4$ and density factor is 15

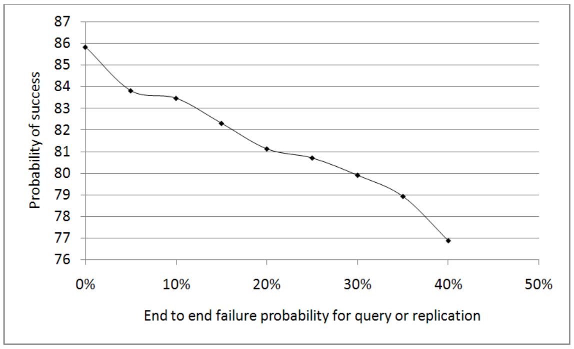

Figure 6. Effect of End-to-End Failure rate on success probability

RDQ$^+$ is also more robust against transmission failures in replicating and retrieving data in the network since both data packets and queries are sent to multiple directions. So, if one of the data packets (readings, tags or keys) or queries fails, there is still a chance to receive the required data from the rest of the replicas. As it is shown in Figure 6, increasing the end-to-end failure probability even as much as 40%, the success probability decreases by less than 10% for the case where $\alpha' = \beta' = 4$ and the density factor is 15.

Regarding the level of privacy achieved, it should be clear that $\mathcal{ADV}$ can find the ID of the queried sensor s only if s belongs to the set of compromised nodes (loss of privacy is proportional to z/N). This is because neither the tagging scheme nor the data stored in replicas reveal any information about the queried node. It is conceivable, however, that in a network where sensors rarely generate any readings, $\mathcal{ADV}$ can locate the queried node if in two (or more) of the replication lines emanating from s, $\mathcal{ADV}$ controls at least two of the nodes in the line. Using location information from the compromised nodes, the queried node would be located near the point of intersection of these two lines.

Assuming each replication line has length $\sqrt{N}$, the probability of having two or more (out of α) replication paths with at least two compromised nodes can be shown to be proportional to $\alpha z^2/N$, for small α and z (details omitted). However, a simple fix exists if we modify the RDQ$^+$ scheme as follows: instead of sending the data along a line, each node forwards the received data to a randomly selected neighbor which makes an angle of more than 90 degrees with the node which it has received the data from. Using this simple mechanism, the privacy problem disappears at the expense of reducing the success probability by a mere 4% (details again omitted).

VII. Conclusions

In this paper, we have considered the problem of query privacy for clients of WSNs in which the network operator cannot be trusted. Without relying on public key cryptography or other heavy cryptographic mechanisms, we have addressed the privacy problem using randomization for getting the required data. Inspired by the data-centric model of communication in sensor networks, we started with the Random Direction Query scheme and gradually improved the performance by providing tradeoffs between storage and communication overheads. All schemes were evaluated both analytically and experimentally, validating their performance and their effectiveness in protecting the sensor IDs to be queried. It is of interest to note that query privacy is achieved using only standard, symmetric cryptography and no knowledge of the topology of the network.

VIII. Acknowledgements

Research leading to these results has been funded by European Community's 7th Framework Programme (FP7/2007-2013), Call reference SEC-2007-1, under Grant Agreement no: 217925.

References

[1] NEPTUNE: http://www.neptune.washington.edu/.

[2] Global Earth Observation System of Systems (GEOSS): http://www.epa.gov/geoss/.

[3] ORION: http://www.oceanleadership.org/programs-and-partnerships/ ocean-observing/.

[4] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensornets," *Computer Communication Review*, vol. 33, no. 1, pp. 137–142, 2003.

[5] Y. Ouyang, Z. Le, Y. Xu, N. Triandopoulos, S. Zhang, J. Ford, and F. Makedon, "Providing anonymity in wireless sensor networks," *Inter. Conf. on Pervasive Services*, 2007.

[6] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "On providing anonymity in wireless sensor networks," *IC-PADS*, 2004.

[7] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[8] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Onion routing," *Commun. ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[9] R. Dingledine, N. Mathewson, and P. F. Syverson, "TOR: The second-generation onion router," in *USENIX Security Symposium*, 2004.

[10] S. Misra and G. Xue, "Efficient anonymity schemes for clustered wireless sensor networks," *IJSNet*, vol. 1, no. 1/2, pp. 50–63, 2006.

[11] E. D. Cristofaro, X. Ding, and G. Tsudik, "Privacy-preserving querying in sensor networks," *CoRR*, 2009.

[12] J. Horey, M. M. Groat, S. Forrest, and F. Esponda, "Anonymous data collection in sensor networks," in *4th Inter. Conf. on Mobile and Ubiquitous Systems*, 2007.

[13] B. Carbunar, Y. Yu, W. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," *TOSN*, vol. 6, no. 2, 2010.

[14] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *FOCS '95*.

[15] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MOBICOM*, 2000.

[16] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in *WSNA*, 2002, pp. 22–31.

[17] H. Solomon, *Geometric probability*. Society for Industrial and Applied Mathematics, Philadelphia, 1978.

[18] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[19] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *ACM Conf. on Computer and Communications Security*, 2002, pp. 41–47.

[20] H. Soroush, M. Salajegheh, and T. Dimitriou, "Providing transparent security services to sensor networks," in *IEEE International Conference on Communications (ICC)*, 2007.