Security in Mobile Wireless Sensor Networks - A Survey

Yi Ren §, Vladimir Oleshchuk §, Frank Y. Li § and Xiaohu Ge ‡
§ Department of Information and Communication Technology, University of Agder (UiA), Grimstad, Norway
‡ Department of Electronics and Information Engineering, Huazhong University of Science and Technology, China
Email: {yi.ren, vladimir.oleshchuk, frank.li}@uia.no; xhge@mail.hust.edu.cn

Abstract— **Thanks to recent advances in robotics, sensors and wireless communications, it is feasible to develop a variety of new architectures for Mobile Wireless Sensor Networks (MWSNs) that play an important role in various applications such as battlefield surveillance, harbor monitoring, etc. However, due to the dynamic of mobile network topology in MWSNs, many new security challenges have emerged. In this article, we give a survey on the state of the art technologies in security aspects of MWSNs. We review existing work that provides security in MWSNs, with an emphasis on data survival, forward secrecy, backward secrecy, authentication, and methods for sensor capture detection. Furthermore, in order to stimulate the exploration of new research areas, we point out a few open research topics that can be further pursued, and also shed light on these topics.**

Index Terms— **Security; Mobile Wireless Sensor Network; Wireless Sensor Network**

I. Introduction

In recent years, Wireless Sensor Networks (WSNs) have been an extremely popular research area [1]. A WSN usually consists of a large number of different types of sensors that are able to monitor a wide variety of ambient conditions such as temperature, humidity, vehicular movement, pressure, noise levels, etc. The low cost of sensors makes it possible to deploy a large number of them to perform both military and civilian applications. However, the low cost of sensors also leads to severe resource constraints such as limited battery power, memory and low computation capability, and these constraints in turn introduce major obstacles to the implementation of traditional computer security approaches (such as public key cryptography) in a WSN. Moreover, the open nature and unattended operation of WSNs make the security defenses even more difficult. Because of these, WSN security issues, such as key management, message authentication, intrusion detection, etc. [2]–[4], have recently gained a lot of attention.

However, despite the static network topology, the traditional WSNs suffer from the following drawbacks:

- Near-Sink sensors drain their energy faster than other sensors in the network because these sensors need to not only deliver their own data to the sink but also forward data originating from many other sensors (located farther away) towards the sink. As a consequence, the near-sink sensors could rapidly deplete their energy and then totally lose their functions.
- Near-Sink sensors would attract more attacks than other sensors do because if enough near-sink sensors are compromised or lose function, sink reachability would be compromised.
- In human being hostile areas, such as battle fields, volcanic areas, underwater zones, etc., sensors are usually deployed by airplanes or helicopters, creating the predicament of imprecise sensor location and coverage uncertainty.
- Moreover, in the abovementioned areas, it may not be feasible to deploy a fixed sink (or base station).

Additionally, we observe that in a large extent of existing WSN security literature (especially in the early research stage), it is assumed that sensors are both static and able to transmit sensed data at will (or anticipate an upload signal) to a trusted party, called a sink or base station, inside the network. However, this is not the case for all real-world WSN applications, for example in certain areas, such as the ocean surface [5] (sensors or sinks are mobile due to water current and wave conditions), underwater [6] (sensors or sinks are mobile because of the water current), patient monitoring [7] and wildlife monitoring [8] (sensors are mobile due to their attachment to patient or alive, moving animals). These new applications therefore require a new network topology in which either sinks or sensors are mobile or both of them are mobile. Thanks to the advances in robotics, it is possible to develop a variety of mobile nodes [9] to form a new type of WSNs known as Mobile Wireless Sensor Networks (MWSNs). In MWSNs, mobile nodes (essentially small robots with sensing, wireless communications, and mobility capabilities) are useful for applications such as adaptive sampling, improving network connectivity of static sensor deployment and event detection. In contrast to conventional static WSNs, targets that might never have been detected in a static WSN can now be detected by mobile sensors/sinks [10] due to sensors' (or sink's) mobility. Finally, mobility enables us

Manuscript received August 15, 2010; revised November 15, 2010; accepted January 15, 2011.
Corresponding author: Xiaohu Ge, Email: xhge@mail.hust.edu.cn

doi:10.4304/jcm.6.2.128-142

to solve network connectivity problems caused by sensor failure or battery depletion. In brief, MWSNs differ from WSN in a way that the network topology of MWSNs is dynamic.

Nevertheless, the unique properties of MWSNs pose many new challenges in security. Due to dynamic mobile network topology, security in MWSNs becomes more complicated. Thus, more exploratory studies are required with regard to the security issues involved. Aiming at providing deeper understanding of current security schemes in MWSNs, we provide in this paper a survey of the existing work in MWSN security that has been proposed in recent years, including a classification for these schemes. Furthermore, in certain topics such as location privacy and authentication, due to an absence of literature addressing security mechanisms in MWSNs, some classic schemes in traditional WSNs are addressed. In order to give a clue on this new research direction, we further highlight a few open questions that need be addressed, together with our suggestions on these topics. It is worth mentioning that this article does not focus on the common security issues of either traditional WSNs or MWSNs, but rather specifically addresses newly arisen issues due to the dynamic mobile network topology of MWSNs. Readers interested in common security issues of traditional WSNs and MWSNs may refer to a few existing surveys on security issues in WSNs [2], [3], [11].

In the remainder of the article, we discuss a number of security challenges unique to MWSNs. In Section II, we summarize the network architecture of MWSNs. The outline of the threat to models of MWSNs is addressed in Section III. The security requirements of MWSNs are listed in Section IV. In Section V, we review the existing schemes on MWSNs. A few new research directions in MWSNs are pointed out in Section VI. Finally, we offer a conclusion in Section VII.

II. Network Architecture

In this section, we classify the network architectures of MWSNs into three-categories: static sensor nodes with mobile sink, static sink with mobile sensor nodes, and situations in which both sensors and sinks are mobile.

A. Static Sensor Nodes with Mobile Sink

A MWSN may consist of a large number of static sensors and a mobile collector (sink or base station). After data collection, the sensors store the data locally till the mobile sink visits the network to collect data. For instance, static sensors deployed in a volcanic area are responsible for collecting parameters of volcano activities. A mobile sink (e.g. a helicopter) periodically visits the network to collect the sampled data.

B. Static Sink with Mobile Sensor Nodes

In contrast, there exists another network topology in which the sink is static while sensors are mobile. Consider, for example, a scenario in which zoologists are interested in monitoring animals' activities and their health status in their natural habitat. They attach sensor devices to animals and then let them roam freely. The sensors thus move with the animals anytime and anywhere, whereas the sinks (or base stations) are deployed in the places where the animals visit frequently, such as water sources for drinking, caves for sleeping, or trees for enjoying the shade. When the sensors attached to the animals generate data, they cannot transmit the data to the sink at will unless they are within the transmission range of a sink. The data generated needs to be stored locally until the animals visit the regions covered by the sinks, and then the sensors attached to the animals can upload data to the sink.

C. Both Sink and Sensor Nodes are Mobile

Besides the aforementioned two MWSN topologies, yet another network topology is that both sink and sensor nodes are mobile. For example, a Sensor Equipped Aquatic (SEA) Swarm [6], consisting of a large number of underwater sensor nodes air-dropped to the chosen venue, was deployed to support applications for time critical applications, such as submarine tracking and harbor monitoring. Each node, consisting of various sensors, a fish-like bladder apparatus and a pressure gauge, can dynamically control the depths using the bladders and on-board pressure gauges [12]. As shown in Fig. 1, a SEA Swarm is deployed in a square region, operating and moving as swarm with the water current, searching for invasive submarine or scouting the waters around harbors or underwater mining facilities. A few unmanned submarines may act as mobile sinks to receive alert messages from sensor nodes.

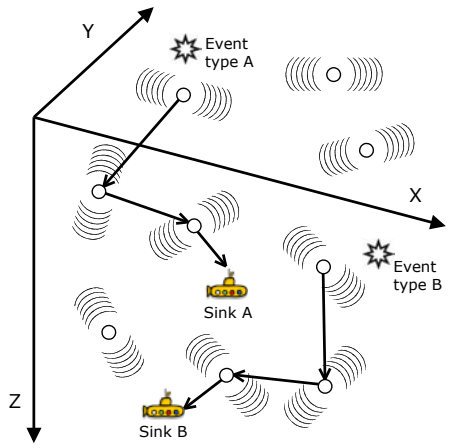

Figure 1. An example of a SEA Swarm: mobile nodes detect events and report them to corresponding mobile sinks.

As discussed above, all these three network topologies of MWSNs have the common property that sensors must be able to store sensed data in their memories until meeting with sinks to upload their data.

III. Threat Model

Due to the fact that sinks (or base stations) are not always present in MWSNs, new security threats emerge.

In this section, we focus on these new challenges which are unique in MWSNs. While we focus more on attacks in MWSNs, interested readers are referred to [11], [13] for more in-depth information on common attacks in WSNs.

A. Outsider Attacks

Because of the open nature of wireless communications, unauthorized participants of the network are able to eavesdrop on the radio frequency of MWSNs. For example, in a battlefield application, sensors are used to sense noise, vibration and light caused by troop movement. An adversary can alter or spoof packets to induce inaccurate analysis results based on the bogus sensed data.

Another outsider attack is to disable the function of sensor nodes. By doing so, an adversary can launch Denial of Service (DoS) attack or inject useless packets to drain the resource of receivers. Moreover, the adversary may physically corrupt sensors (such as smash, melt or corrode them); consequently, the sensors totally lose their functionality.

B. Insider Attacks

Insider attacks happen when authorized sensors of MWSNs behave in unintended or unauthorized ways. When sensors are captured by an adversary, it can reprogramme the sensor and hold the secret key. With the compromised sensor, the adversary can perform insider attacks such as generating bogus data, seeking to steal secrets from the network and disrupting its normal functioning.

C. Mobile Adversary

In MWSNs, sensors cannot transmit sensed data to a sink at will because the sink is not always present (i.e., a mobile sink periodically visits the static sensors to collect data; mobile sensors visit static sinks to offload data; or mobile sinks and mobile sensors meet and communicate uncertainly). The data accumulated in their memories thus become targets of various adversaries.

The authors in [14] propose a mobile adversary model where a mobile adversary visits and travels around the network, trying to compromise a subset (up to a certain size) of sensors within the time interval while sinks are not present in the network. (We will hereafter refer to the mobile adversary as $\mathcal{ADV}$.) The time an $\mathcal{ADV}$ compromises a set of sensors is much shorter than the time between two successive data collections of a sink. As a consequence, given enough compromise intervals, such an $\mathcal{ADV}$ can gradually subvert the entire network.

The authors in [15] further extend and divide an $\mathcal{ADV}$ into three categories as follows:

- *Read-only* $\mathcal{ADV}$: $\mathcal{ADV}$ aims to learn as much sensed data as possible. Until now, it was not difficult to read data from the memories of commodity sensors [16]. Therefore, with no countermeasures, an $\mathcal{ADV}$ can compromise sensors and read the data accumulated in their memory directly.
- *Search-and-erase* $\mathcal{ADV}$: $\mathcal{ADV}$ tries to prevent certain target data from reaching the sink. For example, in a nuclear emission monitoring application, sink will raise the alarm if one of the sensing nodes reports a value above a pre-specified threshold. $\mathcal{ADV}$ thus aims to find that value and erase it before it ever reaches the sink. $\mathcal{ADV}$ might be undetected if the sink tolerates some missing measurements (due to occasional errors or malfunctions).
- *Search-and-replace* $\mathcal{ADV}$: If the sink has no tolerance for lost data, $\mathcal{ADV}$ changes its strategy from search-and-erase to search-and-replace, in the sense that $\mathcal{ADV}$ replaces the target data with a value within the threshold.

D. Node Failure or misbehavior

Unlike traditional WSNs, sensors in MWSNs should be able to store data in their memories till meeting a sink to offload the data. As a consequence, if the sensor node fails or operates incorrectly, all the accumulated data will be lost. Thus, the impacts of node failure in MWSNs are more serious than in traditional WSNs. Node failure can be caused by several reasons such as node compromise, battery depletion and physical damage (i.e., natural disaster or being smashed by an attacker).

IV. Security Requirements

In this section, we introduce security requirements which need to be met in MWSNs.

- *Confidentiality*: Due to the open nature of MWSNs, confidentiality is necessary to enable sensors to protect data from eavesdroppers. For example, a message should be understood by desired recipients rather than attackers. The standard solution to keep sensitive data secret is to encrypt the data with a secret key known only by the desired recipients. Since public key cryptography is too resource demanding for commodity sensors, one can use symmetric key encryption (e.g., Data Encryption Standard (DES) and Advanced Encryption Standard (AES)) and a shared secret key between the communicating parties to achieve confidentiality.
- *Integrity*: Ensuring the integrity of sensed data is important for preventing data from being modified by attackers. Unlike confidentiality, integrity is, in most cases, a mandatory property. Upon receiving a message from a sensor, the sink wants to ensure that the received message is exactly the same as that is sent by the sensor (i.e., it contains no modifications, insertions, deletions, or replays).
- *Authentication*: Authentication defines the security requirement that any sensor node can ensure the identity of the peer node with which it is communicating [17]. Since communication between sensor nodes is based on wireless communication medium, sensor nodes should be able to detect maliciously

injected or spoofed packets. In addition, data integrity and sensor authentication are essential security requirements in most sensor applications [2]. An attacker might modify and delete data generated by sensors. Data generated by sensors thus have to be authenticated before they are accepted as valid data and are used for whatever purposes. To provide data authentication, common approaches are Message Authentication Codes (MACs) or digital signatures, which are usually used in the applications where data authentication and data integrity are needed.

- *Access control*: Ensuring that only authorized users can access the network resource (i.e., the policy controls which can access a resource, under what conditions access can occur, and what those accesses to resources are allowed to do).
- *Availability*: Providing availability requires that a MWSN is always accessible throughout its lifetime. A common way to compromise network availability is DoS attack that involves saturating the network with external communication requests so that it cannot respond to legitimate traffic. In addition, sensor failure caused by battery depletion or hardware error can also compromise availability. In practice, loss of availability may have serious impacts. For instance, during battlefield surveillance, a compromising of availability may open a back door for enemy invasion.
- *Forward Secrecy*: Ensuring forward secrecy requires that an $\mathcal{ADV}$ should not be able to read any previously transmitted message.
- *Backward Secrecy*: Providing backward secrecy requires that an $\mathcal{ADV}$ should not be able to read future messages after it leaves the MWSNs.
- *Auditing*: Ensuring auditing requires that sensors of MWSNs should have the ability to store any significant events that occur inside the network. Auditing is necessary because of the autonomous nature of the sensors. In some scenarios, sink (or base station) will not always be available, so sensors should be able to store the sensed data till a mobile sink visits the network to retrieve the data.
- *Non-repudiation*: Providing non-repudiation requires protection against denial by one of the communication parties. More specifically, a node (or sink) cannot deny sending the message it has previously sent. A common method of providing non-repudiation is to produce certain *evidence* to prove that the communication party has performed a task.
- *Privacy and Anonymity*: Providing privacy and anonymity is very important. In some scenarios, identities and locations of sensors and sink (or base station) should be protected or hidden. For example, when considering battlefield surveillance, soldiers equipped with mobile sinks, which can be used to access the network to obtain information about enemy activities, will be in danger if their locations are revealed.

V. Overview of Existing Security Mechanisms

In this section, we introduce our taxonomy of security mechanisms for MWSNs. As shown in Table I, we classify and review the existing work on security of MWSNs based on security requirements of MWSNs. Generally speaking, there are more security requirements needed in MWSNs (as discussed in Section IV). However, since security requirements (integrity, availability, etc.) are also common requirements in WSNs, existing work [2], [3], [11] proposed for them can be used in MWSNs as well. Here, we discuss security issues unique to MWSNs that arise due to their dynamic network topology. In addition, due to an absence of literature addressing certain areas of security mechanisms, we also introduce some classic schemes in traditional WSNs as an alternative.

TABLE I.
Taxonomy of security mechanisms for MWSNs.

Issues	Publications
Secrecy	[18]–[20]
Data Survival	[14], [21], [22]
Authentication	[23], [24]
Access Control	[25]–[27]
Access Privacy	[27]
Data Source Location Privacy	[28]–[30]
Sink Location Privacy	[31], [32]
Key Management	[33], [34]
Intrusion Detection	[35], [36]
Intrusion Resilience	[37], [38]

A. Secrecy (Countermeasures against Read-only $\mathcal{ADV}$)

To solve the security issues arising due to read-only $\mathcal{ADV}$, we have to guarantee both forward secrecy and backward secrecy. As shown in Fig. 2, let us assume that an $\mathcal{ADV}$ compromises a sensor node s_i at round r_1, and releases the s_i at round $r_2(r_1 < r_2)$. Between round r_1 and r_2, the $\mathcal{ADV}$ is residing in s_i, and we define this time interval as *reside period* T_{rp}. Thus, the forward secrecy is secrecy of data generated before round r_1. The forward secrecy of a sensor s_i is compromised if the data generated and encrypted *before* round r_1 can be decrypted by an $\mathcal{ADV}$ which holds the secret obtained during reside period T_{rp}. In contrast, backward secrecy is about secrecy of data generated after round r_2. The backward secrecy of a sensor s_i is compromised if the data generated and encrypted *after* round r_2 can be decrypted by an $\mathcal{ADV}$ which holds a secret obtained during reside period T_{rp}.

Existing schemes in cryptography were proposed to achieve either forward secrecy [39], [40] only or both forward secrecy and backward secrecy [41]–[44]. Key evolution is a common approach in all of these schemes to provide forward secrecy. Its basic idea is that secret key $\mathcal{K}_i$ is updated by applying hash function at each round, e.g., $\mathcal{K}_i^r = h(\mathcal{K}_i^{r-1})$ $(r \geq 1 \,\text{and}\, \mathcal{K}_i^0 = \mathcal{K}_i)$. Because

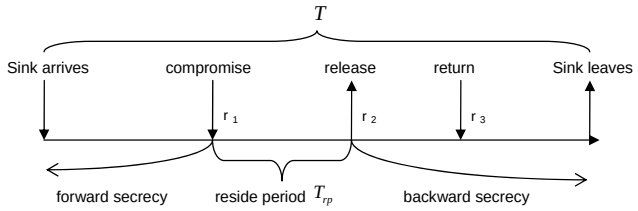

Figure 2. A node is compromised by $\mathcal{ADV}$ at round r_1, released at round r_2, and at round r_3 the $\mathcal{ADV}$ returns again.

of the one-way property of hash function, deriving the previous rounds key (pre-compromised) based on current round key is impossible.

However, the aforementioned public cryptography approaches [41]–[44] are not suitable for MWSNs because sensor nodes use only the public key of the sink to encrypt data. The private key evolved at each round is kept in the mobile sink which is assumed to be a trusted party, whereas sensors are usually considered as source constrained devices that cannot afford for public key cryptography. Therefore, symmetric key-based schemes are needed. In the rest of this subsection, we summarize two popular schemes designed specifically for MWSNs.

DIstributed Self-Healing (DISH) [18]. In DISH, the authors first define a mobile adversary that has compromising capability k, meaning that the mobile adversary can compromise up to k nodes during the time interval T while the mobile sink is not available. Three states of sensors are defined (called *healthy*, *sick*, and *occupied*) in DISH. At the initial stage, all sensors are *healthy*, in the sense that $\mathcal{ADV}$ has no information about their data and secret keys. As soon as the mobile sink collects data and leaves the network, the $\mathcal{ADV}$ starts to compromise sensors. A sensor is *occupied* if it is currently compromised. Once the sensor is compromised, the $\mathcal{ADV}$ can access the local memory and learn the secret key. The sensors are *sick* if the sensors have been previously compromised, and the $\mathcal{ADV}$ can still compute the current secret keys based on the keys acquired when it occupied them, even if it no longer does so.

The basic requirement of DISH is that unattended sensors attempt to recover from sick states and maintain both forward and backward secrecy of the collected data. In DISH, forward secrecy is provided through key evolution. However, DISH does not guarantee absolutely backward secrecy. Instead, it provides probabilistic backward secrecy, a state which depends on conditions such as: compromising capability of the mobile adversary (number of nodes it can compromise at a given time interval T), and for how long time the mobile sink successively visits the network.

The main idea of DISH is to let healthy sensors heal *sick* sensors. As shown in Fig. 3, sensor A receives contribution r from its neighbors B, C and D, which is a random number generated by its corresponding neighbors, and then uses the contributions along with its current key as input to a one-way hash function to compute its next round key. If at least one of the sensors B, C, D is healthy (in the sense that at least one of the contributions r_B, r_C, r_D is unknown to the $\mathcal{ADV}$), the $\mathcal{ADV}$ is unable to learn the newly generated key. Here, the contribution is a pseudo-random value. As a consequence, sensor A is *healed*, and its state transfer from sick to healthy. We claim that DISH can provide probabilistic backward secrecy because if all its neighbors are sick, sensor A cannot become healthy. Fig. 4 further shows the states transition of sensors. With the sensor cooperation approach, healthy sensors always remain healthy as long as they are not directly compromised, whereas sick sensors can become healthy if at least one of their received contributions is from healthy peers.

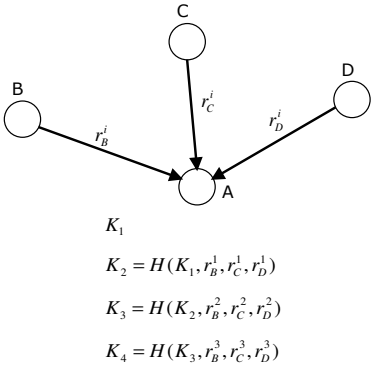

Figure 3. An example of DISH.

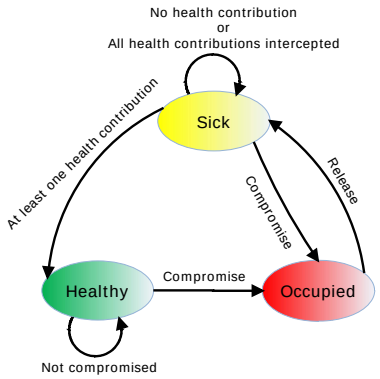

Figure 4. DISH sensor states transition diagram.

Proactive co-Operative Self-Healing (POSH) [19]. Compared with DISH [18], the basic idea of self-healing in POSH is the same, which causes sick sensors receiving random contributions from healthy peers to become healthy again. The difference is that the DISH scheme is referred to as a *pull* model because sensors request contributions from peers; on the contrary, the POSH scheme is referred to as a *push* model that involves sponsors volunteering their contributions. In other words, in the *push* model, sensors send their contributions to t randomly selected neighbors without receiving any requests.

POSH improves DISH in the following significant manner: the POSH scheme can guarantee better probabilistic backward secrecy than the DISH scheme. In the DISH scheme, $\mathcal{ADV}$ knows the peers from which each sick

sensor has asked for a contribution. For example, as shown in Fig. 3, sensor A requests contributions from its neighbors B, C, and D. If sensor A is sick, meaning that it was compromised before, the $\mathcal{ADV}$ knows its previous key $\mathcal{K}_{r_1}$ and the set of sensor A's future sponsors. The $\mathcal{ADV}$ can then compromise all the sponsors in the set and acquire their contributions to sensor A, which allows $\mathcal{ADV}$ to compute the new key. However, in the POSH scheme, the sponsors randomly select t neighbors to send contributions. As a consequence, if a sponsor is healthy, $\mathcal{ADV}$ cannot determine for certain the set of sensors it might contribute to, meaning that the $\mathcal{ADV}$ cannot know all the sponsors who sponsor the contributions to sensor A's new key. Moreover, the POSH scheme requires only half of the message required in the DISH scheme because DISH needs two messages - a request and a reply - for each contribution.

More recent work [20] utilizes (m, n) Reed-Solomon (RS) coding to improve data reliability and backward secrecy. Sensors take advantage of (m, n) RS codes to divide data into n parts and then send parts of their data to their neighbors, where m of n ($m < n$) data parts are required to reconstruct data, adding data redundancy to provide resilience to node invalidation and Byzantine failure. Data can be recovered if less than $n - m$ data parts are compromised. Since data parts are distributed among sensor's neighbors, backward secrecy of s_i can be compromised if and only if

1) s_i is compromised by the mobile adversary.
2) The mobile adversary's compromising ability is $\theta > m$.
3) The mobile adversary has compromised at least m neighbor nodes of s_i that store the corresponding data parts.

Our scheme in [20] makes an obstacle that forces the mobile adversary to compromise more sensors to compromise s_i's backward secrecy. Assuming that P_i is the probability that s_i can be compromised by the mobile adversary, and that $P_{i,j}$ is the probability that $s_{i,j}$, a neighbor node of s_i, is compromised, it is easy to understand that the backward secrecy of s_i is improved due to the fact

$$P_i \Pi_{j=1}^{m} P_{i,j} < P_i \,.$$

$P_{i,j}$ is usually different for different nodes and could be evaluated from the feedback of certain security monitoring software and/or assigned manually by the mobile sink based on information such as the physical protection, the location, or the role of the nodes. We further improved the scheme [20] by selecting the top m security level nodes, e.g., m pairs with the lowest $P_{i,j}$ to decrease $P_i \Pi_{j=1}^{m} P_{i,j}$.

B. Data Survival (Countermeasures against Search-and-Erase $\mathcal{ADV}$)

Let us consider a scenario where a network of nuclear emission sensors are deployed in a recalcitrant country (under an international treaty) in order to monitor any potential nuclear activity [14]. If one of the sensors senses a value above a certain threshold, the sink can raise an alarm to report the emission. An adversary might want to find that value and erase it before it reaches the sink. This adversary, as discussed in Section III-C, is called Search-and-Erase $\mathcal{ADV}$. If the network can tolerate some missing measurements (due to message failure or lost communication), the Search-and-Erase $\mathcal{ADV}$ remains undetected even if it succeeds.

In [14], later extended in [21], the authors target a Search-and-Erase $\mathcal{ADV}$ to propose countermeasures. Recall that the main threat of a Search-and-Erase $\mathcal{ADV}$ is to prevent certain target data from reaching the sink. In other words, the idea of countermeasure can be simply described as how to keep the certain target data survival. Based on the idea described above, [14] proposed three data survival strategies: DO-NOTHING, MOVE-ONCE, and KEEP-MOVING.

- DO-NOTHING. In this strategy, sensors do nothing about their sensed data. That is sensors just simply leave the data to reside in the sensor that collected it and wait for the sink. In this case, the Search-and-Erase $\mathcal{ADV}$ can find the target sensor quickly and erase the target data.
- MOVE-ONCE. In this strategy, sensors move their data once and only once right after their data collection. The data will stay in their new *home* until the next sink visits.
- KEEP-MOVING. Sensors move data continuously, i.e., at each round, each sensor re-allocates the data to another randomly selected sensor.

To analyze data survival probability, the authors have further proposed three attack strategies of the Search-and-Erase $\mathcal{ADV}$ as follows:

- LAZY. The $\mathcal{ADV}$ is LAZY in the sense that the $\mathcal{ADV}$ satisfies with its current compromised k sensors, and does not want to further compromise other k sensors.
- FRANTIC. It is only in extreme cases that $\mathcal{ADV}$ compromises k randomly selected sensors at each round.
- SMART. $\mathcal{ADV}$ first selects two sets of sensors, each of size k, and then simply alternates control between these two sets at each round.

At the first glance, we may have the impression that DO-NOTHING is the worst choice listed above, and KEEP-MOVING is the best one to keep target data survival since it is more difficult for $\mathcal{ADV}$ to catch the target data if it is moving continuously. However, this supposition is not true. After completing a detailed analysis comparing the survival strategies of target data and attack strategies of $\mathcal{ADV}$, the authors have produced the analysis results (as shown in Table II). It is surprising that DO-NOTHING is the best choice if $\mathcal{ADV}$'s attack strategy is LAZY and SMART. For the $\mathcal{ADV}$, a "NO" label in a table cell implies that the corresponding combination of

data survival and attack strategies is not sensible, whereas a "YES" label means that the corresponding combination is viable.

TABLE II.
VIABILITY OF SURVIVAL AND ATTACK STRATEGIES [14].

	Attack Strategy		
Survival Strategy	LAZY	FRANTIC	SMART
DO-NOTHING	NO	YES	NO
MOVE ONCE	NO	YES	NO
KEEP MOVING	YES	YES	YES

Although sensors may play a hide-and-seek game by moving all data around the network [14], [21], [22], this is ultimately a losing game, unless cryptography is used [22]. The authors in [22] apply cryptography in the context of data survival in unattended WSNs. By comparing with symmetric encryption and public key encryption, the result shows that simple cryptographic schemes coupled with data mobility strategies can be of great help in providing data survival, and demonstrates that there is no security advantage in using public key (over symmetric) cryptography.

C. Authentication (Countermeasures against Search-and-Replace $\mathcal{ADV}$)

In traditional WSNs, schemes [45]–[48] were proposed to provide data (sensor) authentication. However asymmetric cryptographic mechanisms are too costly for resource constrained sensor networks. To address this problem, the authors in [45] proposed the μTESLA protocol that introduces asymmetry through a delayed disclosure of symmetric keys and one-way function key chains, resulting in an efficient broadcast authentication scheme. In [46] authors improved the μTESLA key distribution efficiency by using predetermined and broadcast initial parameters of the μTESLA instead of unicast-based initial key chain distribution, thus the requirement of unicast-based initial communication between sensor nodes and base station to save communication overhead in large-scale WSNs is removed. This is because if a node wants to broadcast information in the μTESLA, it must first firstly send the information to the base station, and then the base station broadcasts the information. This procedure causes in turn a great deal of communication overhead between base station and sensor nodes and is not suitable for local broadcast authentication. The authors in [48], [49] have proposed an efficient protocol for providing local broadcast authentication based on the use of one-way key chains to solve the problems in the μTESLA. To prevent false data injection attacks in WSNs, the authors in [47] enable the base station to verify authenticity of a received report as long as the number of compromised nodes is fewer than a certain threshold value.

However, the schemes mentioned above are not suitable for MWSNs since sink (or base station) is not always present in this case. Furthermore, due to the absence of real-time communication between sensors and sinks, sensors are forced to accumulate data along with their authentication information till the next visit of a mobile sink to offload the data. The authentication information per data is not a storage issue in traditional WSNs because sensors can send data alone with the authentication information to a base station either at will or at brief intervals, which would not cause much storage overhead, whereas in MWSNs, computing authentication tags per sensed unit of data may cause high storage overhead, and the cost increases along with the mobile sink visiting time interval. Moreover, in order to provide reasonable security requirements, a minimum number like 128 bits per MAC (or 320 bits per signature) is needed. Consequently, the size of the authentication tag may easily exceed the size of sensed data. Thus, how to solve the storage overhead of authentication accumulated tag is a challenging task in MWSNs. In addition, if a sensor is compromised, its secret key used for MACs or signatures is exposed as well. An $\mathcal{ADV}$ holding a secret key can easily produce falsely sensed data *after the compromise*, and also produce fraudulent data *before the compromise*, if the data has not been off-loaded to a sink. Therefore, another issue is how to provide forward secrecy.

To reduce the storage overhead of authentication tags as well as provide data (or sensor) authentication and forward secrecy, the authors in [23] explore Forward secure sequential Aggregate (*FssAgg*) authentication schemes [50]–[53] to construct an FssAgg-MAC scheme and an FssAgg-signature scheme, which allows a signer to combine multiple authentication tags generated in different key/time periods into a single constant-size tag. Consequently, the storage overhead of accumulated authentication tags is reduced significantly. However, although computational efficiency is achieved through hash chains and symmetric key distribution, FssAgg-MAC still requires high storage overhead, and its signature cannot be publicly verified.

To achieve computational efficiency, forward secrecy and public verifiability, the authors in [24] proposed a new class of digital signature schemes for MWSNs, which is called Hash-based Sequential Aggregate and Forward Secure Signatures (HaSAFSS) by introducing asymmetry between the senders (sensors) and receivers (mobile sinks) with the aid of Timed-Release Encryption (TRE) [54]. The property of TRE is to encrypt data that no party including the intended receivers (mobile sinks) can decrypt it until a predefined future time. Thus, even if the sender (sensor) is compromised, such property can still provide forward secrecy and aggregate signature in a publicly verifiable way. Based on the difference of keys (time trapdoor key, per-interval key and per-data item key) generation, HaSAFSS can be further classified into two schemes, a symmetric HaSAFSS scheme (Sym-HaSAFSS) and an Elliptic Curve Cryptography (ECC)-based HaSAFSS scheme. Sym-HaSAFSS and ECC-HaSAFSS are complementary to each other with respect to storage overhead of computing these keys. In Sym-HaSAFSS, each sender

(sensor) initially stores R encrypted keys, which each receiver (mobile sink) stores only one key. In contrast, in ECC-HaSAFSS, each sender stores only one key and the sender can compute its own session keys after the deployment based on the fact that each receiver stores R public keys for each sender.

D. Access Control

Access control is defined as the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner [17]. Generally, the data access strategy of WSNs can be divided into two ways, namely access control at the base station and access control at sensor nodes. In traditional WSNs, sensor nodes transmit generated data to a base station that users can access by querying through the base station. Access control strategy is adopted in the base station side rather than in the sensor nodes. Since the base station is not constrained by resources (computation, energy, memory, etc.), many access control policies [55], [56] can be used. As this paper focuses mainly on MWSNs, we do not explicitly address the access control strategies of base stations. In MWSNs, as a large amount of sensed data are stored in individual sensor nodes, the data storage and access have to be protected by using encryption, so that the data can only be accessed by authorized users with the corresponding keys.

For example, in mission-critical application scenarios (such as a battlefield), different kinds of sensors generate different types of sensed data (e.g., smoke, vibration or noise, etc.), which may be specified to different kinds of users (e.g., scouts, landmine experts or officers, etc.), or can be accessed based on the user's security level. For example, a general may have a higher privilege to access data than a soldier does.

To solve the aforementioned problem, related schemes [25]–[27] have been proposed, and they can be divided into two categories, namely Symmetric Key Cryptography (SKC) based schemes [25] and Public Key Cryptography (PKC) based ones [26], [27].

In the SKC-based schemes, data is encrypted and decrypted by using the same secret key. This means that if a sensor is compromised, the $\mathcal{ADV}$ can get the secret key stored in the sensor memory and is thus able to decrypt the accumulated data generated by the same sensor. To solve this problem, a naive scheme is to divide the lifetime of sensors into a series of phases, and the key used for encrypting the data is updated in each phase. Sensors merely store the key for the current phase, and erase all the previous keys securely. Nevertheless, the interaction may incur high computation overhead. The authors in [25] have proposed an SKC-based distributed data storage and retrieval scheme, where the access control policy is provided by sharing the symmetric secret key with authorized users based on perturbed polynomial technique [57]. However, the scheme [25] was proved as not secure in [58]. Thus, distributed data storage access control in MWSNs using SKC-based scheme is still an open question.

Next, in the PKC-based schemes, data is encrypted by using a public key, and it can only be decrypted by using the corresponding private key. Thanks to this advantageous feature of PKC, an attacker is not able to decrypt the data stored in the sensors even if the sensors are compromised, since it lacks the corresponding private keys. In traditional PKC-based schemes, however, data encrypted using a public key can be only decrypted by using the corresponding private key, in the sense of a one-to-one relationship. For instance, the encrypted data using a general's public key can only be accessed by the general. If the data is specified to be accessed by users having different security levels (e.g., scouts or landmine experts), the data has to be encrypted by the public keys of the users, which consequently causes large storage overhead and computation overhead. To tackle this problem, the authors in [26] proposed Fine-grained Distributed data Access Control (FDAC) scheme based on Key-Policy Attribute-Based Encryption (KP-ABE) [59].

The basic idea of KP-ABE is that ciphertext associates with a set of attributes, and private key associates with an access structure. A ciphertext can be decrypted only if the attributes of the ciphertext satisfy the access structure of the private key. As shown in Fig. 5, a landmine expert with the key which has access structure as {(Location is road) AND (Type is smoke OR vibration) AND (Owner is landmine expert)} can decrypt ciphertext *A* with attribute {location = road, type = vibration, owner = (landmine expert, scout, general)}, but cannot decrypt ciphertext *B* with attribute {location = village, type = smoke, owner = (scout, general)}.

By using the advantageous property of KP-ABE, in FDAC, each sensor and its sensed data are associated with a set of attributes, and each user is assigned with an access structure which is embedded in its secret key. The leaf node of the access structure tree is one of the attributes in the attribute set. It is easy to observe that the definition of the access structure tree enables one to represent complex logical expressions, and is able to specify access privileges of users in a fine-grained way. Although the authors have tailored and adapted KP-ABE for WSNs and demonstrate that FDAC is affordable compared with high-end sensor nodes such as iMote2, the scheme is still too expensive for normal sensor nodes because KP-ABE is public key cryptography.

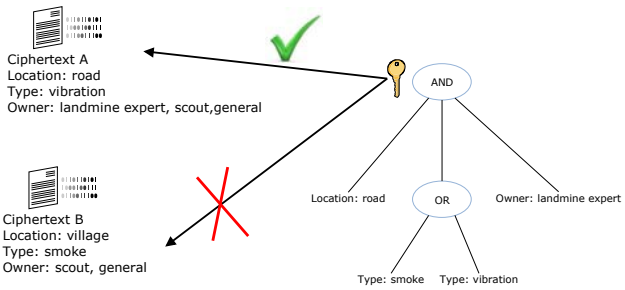

Figure 5. A example of access structure in a battlefield scenario.

E. Access Privacy

In some scenarios (such as ORION [60], NOPP [61] and IOOS [62]), owners and users of the sensor networks are different. To compensate for the cost of operating and maintenance, the owner of a network may enforce access control that only authorized users can access the sensed data. In particular, users may want to keep their privacy when accessing the sensed data in which they are interested, or from which nodes the data is obtained. For instance, an oil company interested in the data of an ocean sensor network [60]–[62] may want to hide its network regions of interest from both the owner and other users of the network (who might be potential business competitors [63]). To guarantee access privacy of network users, the authors in [63] have proposed target-region mix schemes such as uniform, randomized and hybrid transformation, which mix the target-region of the query based on pre-defined mix functions. The basic idea is to hide the target-region by mapping the target-region into k regions, such that the target-region is hided in the k regions where the owner and other users of the network cannot distinguish the target-region from the other uninteresting regions.

In the Union Transform (UT), the target-region of query is transformed into a set of regions. Then, the set of regions is sent to the based station instead of the target-region. The base station answers the data of region in the set. Consequently, it only knows that the target-region of the user is included in the set, but cannot distinguish it from other regions. For example, assuming that $\mathcal{R} = (R_1, R_2, \cdots, R_k)$ is a set of regions, the whole set of regions is sent to the base station to query data if the user wants to query the data from one region $R_i \in \mathcal{R}$.

In the Randomized Transform (RT), the user also sends a set of regions to the base station, instead of the target-region. However, the difference is that the set which consists of regions is selected randomly rather than pre-defined in UT. The Hybrid Transform (HT) is a combination of UT and RT.

To the best of our knowledge, [63] is the first publication that takes into consideration the access privacy of network users in WSNs. However, the scenarios where network users query data via one or several base stations do not exist in MWSNs and thus the scheme proposed in [63] is not suitable for MWSNs.

The authors in [27] have proposed a Distributed Privacy-Preserving Access Control (DP^2AC) scheme that considers a MWSN scenario where the base station may or may be not available. DP^2AC enables to disconnect the mapping between a user's identity and the query of the user by using the blind signature [64]. A user who wants to access the network needs to buy some tokens from the network owner with blind signatures. If the user wants to access the network, she needs to send the pre-purchased tokens to the network for obtaining the access rights. Let us consider Alice as an example. After purchasing tokens from the network owner, Alice can query data from any sensor node, denoted as node A. When node A receives a token, it first checks whether the token is valid, then grants the access right to Alice if it is valid. Moreover, since the validation of the token is verified by checking the signature of the network owner, the token can still be checked as a valid token if it is used before.

To prevent token reuse, the authors in [27] have also proposed a group of four schemes for Token-Reuse Detection (TRD). They begin by describing a network-wide flooding scheme called Scheme 1 in their paper in which each node is its own witness and records all the tokens that have been used by itself or by others. To reduce large storage overhead, each node has to store a replication of the used token, and to reduce large communication overhead caused by flooding, a Randomized Mapping scheme (called Scheme 2) has been proposed. In this scheme, upon receiving a token $\mathcal{T}$, node A sends a TRD request including $\mathcal{T}$ to β witness selected nodes using a geographic routing scheme such as GPSR [65]. Since the results of Scheme 2 are unsatisfactory, they have proposed a Randomized Mapping Plus (called Scheme 3), which allows each node within the transmission range of its forwarding path and thus can overhear the TRD request sent by node A, to return a TR alarm to A if it stores $\mathcal{T}$. Compared with Scheme 2, the TRD probability of Scheme 3 is improved significantly for the same β without additional storage cost.

However, with regard to both Scheme 2 and Scheme 3, there is a tradeoff between the TRD probability and storage overhead as well as communication overhead. The larger the β, the higher the TDR probability and the larger storage and communication overhead. Thus, they have further proposed a Double Ruling (DR) scheme without the aforementioned limitations (called Scheme 4), which is motivated by the DR [66] techniques. The basic idea of DR is storing the sensed data along a continuous curve, called *replication curve* that follows the horizontal line, instead of storing data replication in one or several isolated nodes. As regards network users, they query data along the other continuous curve (called *query curve*) that follows a vertical line. The data can be retrieved if the two curves intersect. Let us consider a simple case. Assume the network is a two-dimensional grid (see Fig. 6). The token storage curves follow the horizontal lines (red/dark line). The TRD request travels alone the query curves that follow the vertical lines (blue/gray line). Suppose that Alice wants to reuse token $\mathcal{T}$ at node A. Upon receiving $\mathcal{T}$, node A sends a TRD request traveling alone the blue line in both up and down directions. When the query curve hits the replication curve, the intersect node can send a TR alarm to node A. Consequently, Alice fails to reuse token $\mathcal{T}$.

F. Location Privacy

Since WSNs are usually deployed in unattended areas, location privacy is an important issue. Location privacy in MWSNs may be classified into two categories: location privacy of sensor nodes (e.g., data source) and location privacy of mobile sinks.

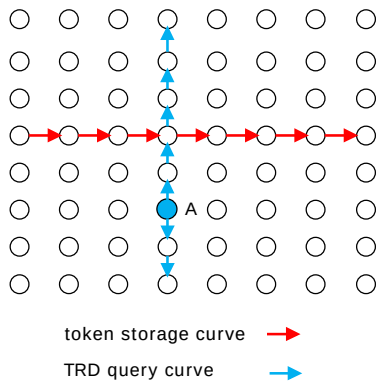

Figure 6. A simple double ruling scheme on a grid.

For location privacy of a data source, consider, for example, "Panda Hunter Game", where a WSN consists of a large number of *Panda-detecting-sensors* deployed in a panda habitat [28]. If a sensor detects panda activity, it generates a message and sends this message towards the based station. Meanwhile, due to the open nature of WSNs, an armed panda hunter may listen in and trace location of the sensor that generates the message of panda location.

Compared with location privacy of data source, location privacy of a mobile sink is more important. Imagine a MWSN deployed in unattended areas, such as a battle-field, where the mobile sink is carried by a soldier or a tank in order that they can access or retrieve sensed data to analyze activities of enemies or movement of troops. If the location of the mobile sink is exposed to the adversary, the soldier or tank will be in great danger. Moreover, since a mobile sink always holds authentication keys and pairwise keys of the network, the entire MWSN becomes useless if the mobile sink is destroyed or controlled by an adversary.

In existing literature, there are several ways to trace the location of data source and mobile sink. Here, we present two main approaches. Since both data source and mobile sink need to receive messages, an adversary can trace the location of a data source or mobile sink by analyzing the traffic rate. This *traffic-analysis attack* is introduced and studied in [67], based on the basic observation that near-sink nodes forward more packets than the sensors further away from the sink. By analyzing the packets sent at various locations in the network, an adversary is able to compute the traffic rates (intensities) at these locations, and then estimates the direction of the sink (the denser it is, the closer to the sink). Another type of attack is that an adversary can trace the location of data source or mobile sink by following the movement of packets. This *packet-tracing attack* is first addressed in [28], where the sender's location privacy (instead of the receiver's) is considered. In this attack, by eavesdropping on the traffic, the adversary is able to perform a hop-by-hop trace toward the original data source. In the following paragraphs, we discuss four existing defense measures (*flooding* [28], *Random walk* [28], *dummy injection* [28], [29], [68], and *fake data source* [30]) against these two types of attacks.

1) Location Privacy of Data Source:

Baseline and probabilistic flooding [28]. The basic idea of baseline flooding is that an intermediate node, once it receives a message, broadcasts the message to all its neighbors. As a consequence, all the sensors in the network participate in forwarding messages so that it is hard for an adversary to trace a transmission route back to the data source. In probabilistic flooding, each sensor forwards message it receives with a pre-specified probability, instead of all sensors participating in forwarding messages. Compared with baseline flooding, probabilistic flooding can save energy for the network.

Random walk [28]. Random walk, as its name suggests, performs a few steps of random walk from data source to protect the original one. If an adversary follows the path of transmission back to the data source, it will only be able to figure out the terminal node of random walk.

Dummy data injection. The basic idea of dummy data injection is to perturb traffic by sending fake packets. As a consequence, the traffic observed by adversary is added with "noise" so that it is not easy to quickly identify fake paths and eliminate them from consideration. In [28], a simple scheme, called *Short-lived Fake Source Routing*, was proposed to protect location privacy. Take communication overhead and energy consumption into account. On the one hand, the scheme enables each sensor to send fake packets with a pre-determined probability. On the other hand, upon receiving a fake packet, a sensor just discards it to prevent from further sending it to others. Although the scheme can perturb the local traffic observed by adversary, it is ineffective when it meets a global adversary that can monitor transmission rate of each sensor node and thus identify the sensors that only sends out dummy packets [29]. To solve the problem, one scheme was proposed in [29] that injects dummy packets globally and keeps the transmission of real packets the same as dummy packets against global adversaries who may monitor and analyze the traffic over the whole network. The main idea of [29] is that sensors send out network-wide dummy packets with intervals following a special distribution, such as constant or probabilistic. Thanks to the fact that sending packets follow a pre-determined distribution, the transmission delay of real packets can be reduced without allowing an adversary to identify the real traffic. However, this benefit is achieved at a cost of transmission power for sensors or in other words shorter network lifetime.

Fake data source [30]. The basic idea of fake data source is to use it to confuse an adversary so that real data source can be protected. By specifying one or more sensors to impersonate the behaviors of real data sources (such as sending fake packets with the same interval or distribution), *more* data sources exist in the network. The more the fake data sources, the harder and slower the adversary can identify the real data source. However, the scheme may cause more energy consumption for the

behavior impersonation.

2) Location Privacy of Mobile Sink:

Besides protecting location privacy of data source, another important task is to protect location privacy of the mobile sink. To the best of our knowledge, there is so far no such literature addressing the topic of protecting location privacy of a *mobile* sink. However, in order to provide a clue and stimulate new research direction, we review the existing approaches aimed at protecting location privacy of a static sink or base station.

The main challenges for protecting the location of a sink or base station can be divided into two types: First, a local adversary may be able to deduce the parent-child relationship according to the received and sent data at each sensor following a certain interval. By analyzing the traffic, the adversary is able to trace down to the sink along the data transmission route. Second, a global adversary is able to monitor all traffic transmitted in a whole WSN so that the global adversary is able to compute the transmission rate of each sensor node, and thus can identify the location of the sink.

Location-Privacy Routing protocol (LPR): In [31], the behavior of an adversary is defined, which trusts previous packet movement trend more than the current eavesdropping trend. To defend against such an attack pattern, the scheme enables sensors to select routing paths randomly based on a pre-defined probability. More specifically, the neighbors of each sensor are divided into two lists - closer list and farther list - based on the hop account from the base station. When a sensor forwards a packet, it selects the next hop from the farther list with probability Pr, and from the closer list with probability $1 - Pr$. Consequently, various routing paths are produced, which in turn drastically reduce the probability of successful analysis by the adversary. In addition, the authors combine the routing protocol with *fake packet injection* to minimize the information that an adversary can deduce from the eavesdropped packets about the direction towards the sink.

Controlling transmission rate: Since near-sink sensors need to both generate and forward traffic and thus cause a higher transmission rate (and thus cause a high transmission rate), the asymmetric traffic flow enables an global adversary to find the sink. To defend against the attack, ref. [32] has proposed a privacy-preserving technique to keep the same transmission rate among all sensors by controlling delay of real data.

G. Key Management

In this subsection, we review the existing approaches to hiding the location of mobile sinks in order to prevent mobile sinks from becoming compromised by adversaries. Indeed, once a mobile sink is compromised, the granted privileges to the mobile sink can be abused. On the one hand, without suitable restrictions, a compromised mobile sink will be able to collect data from sensors in the entire network. Since commodity sensors are always constrained by their memory sizes and have to delete the data offloaded to the mobile sink, the data consequently collected by the compromised mobile sink will be lost and cannot be recollected. On the other hand, we may request that mobile sink revoke or isolate a sensor if the sensor is identified as compromised, in order to investigate an abnormal area of a sensor network when we suspect that some sensors in that area may be compromised. Again, without appropriate restrictions, a compromised mobile sink can easily revoke any sensors at will and *freeze* the whole network by simply sending revocation messages. The severe consequence of mobile compromised sinks can also be foreseen in other applications. This consequence exhibits the importance of restricting the privilege of mobile sinks [33].

As discussed above, if a mobile sink is given too high privileges, it will become as the target for attacks and probably compromised. Therefore, security mechanisms that can tolerate mobile sink compromises are needed. In [33], extended in [34], the authors have first proposed several efficient schemes to restrict the privilege of a mobile sink without impeding its capability of carrying out authorized operations for an assigned task. The basic operation is that each mobile sink takes an authenticator for each task it may enforce. To prevent the authenticator from revealing information due to mobile sink compromises, the privileges of the authenticator are restricted by adding parameters, such as the starting time and the ending time of a task, the type of a task and ID of the mobile sink. Each sensor can authenticate the authenticator of the mobile sink before performing the desired task. Once the mobile sink is compromised, the compromised authenticator can only use for the pre-specified interval [stating time, ending time], for accessing the data related to the pre-specified task. In addition, if the mobile sink is compromised, the base station can use authenticated broadcasts to revoke the authenticator, including the ID of the mobile sink.

H. Intrusion Detection

In the subsection, we review the approaches on how to reduce impact - even if a mobile sink is compromised by an adversary. We introduce the manner of detecting intrusion to sensors first. To compromise sensors, an adversary might capture (e.g., remove) sensors from the network, then re-programme them. Indeed, the adversary might capture a sensor to compromise its secret key, or to re-programme it with malicious code before being able to launch various attacks such as generating bogus data for confusing the analysis results based on the data, or performing Sybil attack [69] where a single node illegitimately claims multiple identities stolen from previously captured sensors. Thus, learning how to detect the sensor capture as early as possible is an important step.

Due to the intrinsic features of MWSNs such as absence of a trusted third party, most existing schemes which rely on a trusted third party are not suitable for MWSNs. The authors in [35] have proposed a scheme depending on cooperation of the honest sensors, without a third party, to detect possible sensor captures. Particularly,

when a(n) (honest) sensor *meets* other sensors located in the network, it can gather *meet* time information about the presence of the sensor it has met. The *meet* time information can thereby be considered as evidence that the sensor exists at the time. If the sensor s_a does not *meet* a sensor s_i for a pre-specified interval λ, sensor s_a will broadcast an alert message suspecting that the sensor s_i is probably captured.

Sometimes, sensor s_a would broadcast a false message claiming that s_i is probably captured, while s_i is still working properly in the network. Since the interval λ is pre-specified, prolonging λ can reduce the probability of false alert messages while at the same time prolong the time to detect the sensor capture. Note that we want to reduce the probability of false alert messages, and we also want to detect the sensor capture as early as possible. Thus, the tradeoff is between the probability of false alert messages and the time to detect the sensor capture. To solve the dilemma, a sensor cooperation-based scheme was proposed [35]. With this scheme, two (honest) nodes can exchange *meet* time information about other nodes which they have met during the past interval, every time when these two trusted nodes meet again. For example, sensors s_a and s_b are responsible for tracking sets of sensors S_a and S_b, respectively. They can exchange *meet* time information of $S_a \cap S_b$ when they meet. If both s_a and s_b are tracking a third sensor s_i, s_a and s_b can compare the time when they last met s_i and update the time which is more recent.

I. Intrusion Resilience

We now turn to the area of intrusion resilience in MWSNs. It is worth mentioning that in this subsection we regard intrusion resilience as that sensors exhibit resistance to attacks by themselves, rather than the other popular intrusion resilient approaches which rely on a trusted third party.

In MWSN, while sensors may move according to certain common mobility models (e.g., random walk) within an interest area, the mobile adversary can just occupy a certain area (we will hereafter call it "corruption area") and wait for sensors to come across there. Once a sensor moves to the corruption area, the mobile adversary can compromise its security. As a consequence, it is difficult for the sensor to regain security, i.e., to obtain intrusion-resilience. To solve this problem, the authors in [37] proposed a sensor cooperation protocol that allows compromised sensors to recover their secure state after being compromised. In brief, sensors take advantage of mobility and cooperation with peers to regain security even after having been compromised by a read-only adversary which aims to learn as much data as possible.

The basic idea of the sensor cooperation protocol (more detailed information can be found in subsection V-A.) is similar with [18], [19] that sensors receive contributions from their neighbors and use those contributions, along with its current secret state, to compute the next round secret. To distinguish the states of sensors, the sensor sets are divided into three distinct groups: *red*, *yellow* and *green*, defined as follows:

- *Red*: the state of a sensor is red if it is currently within the corruption area.
- *Yellow*: a sensor is yellow if it is not in the corruption area anymore, but the mobile adversary still knows its state (i.e., secret key).
- *Green*: a sensor is green if it has never visited the corruption area, or it was red (or yellow), but is now *healed* by the cooperation of its neighbors.

Let us give an example to explain how a sensor, say s_i, in *Yellow* state can be transformed into *Green*. At round $r+1$, s_i moves out of the corruption area, but its secret key $\mathcal{K}_i^r$ is still known by the mobile adversary. For simplicity, we assume that s_i only receives three contributions ϕ_j^{r+1}, ϕ_{j+1}^{r+1} and ϕ_{j+2}^{r+1} from three neighbors s_j, s_{j+1} and s_{j+2}. Then s_i computes its new key using those contributions along with its current key as

$$\mathcal{K}_i^{r+1} = H(\mathcal{K}_i^r || \phi_j^{r+1} || \phi_{j+1}^{r+1} || \phi_{j+2}^{r+1}).$$

Thus, it is easy to conclude that the healing is successful if at least one of the three contributions is unknown by the mobile adversary that at least one of three neighbors is green.

However, the adversary considered in [37] is static and *passive*. It controls a fixed portion of the network deployment area and compromises all sensors moving within it. As a further step based on [37], the authors in [38] investigate another envisaged scenario where an *active* adversary exists. It is *active* in that it chooses the portion of the deployment area to compromise at each round and aim to compromise various sets of sensors per round. Generally speaking, [38] is also based on sensor collaborative protocol addressed in [37].

To summarize, in this section we have classified and surveyed various security mechanisms proposed so far for MWSNs. In the next section, we identify a few relevant open questions for future research in this field.

VI. Open Questions

Although many facets of security issues have been studied, there are still a number of open questions in MWSNs which need to be addressed, as outlined below.

A. Long-lived MWSNs

Let us consider network topology of static sensors with a mobile sink. In this type of MWSNs, while there is no static sink available, a mobile sink visits the static sensors with irregular and even unpredictable frequency. Consequently, each sensor must accumulate sensed data and have the ability to wait *long enough* until a specified signal sent by the mobile sink to offload data onto it. Since the memory size of sensor nodes is limited, no matter how the data are compressed, the memory would be full after *certain* phases. Thus the mobile sink has to access the network to offload data at a reasonable interval. Otherwise, newly collected data (or old data) would be lost.

However, in realistic scenarios, the mobile sink may fail to visit the UWSN as planed for any unpredictable reasons (bad weather or blocked by adversaries). In addition, the mobile sink may be specified to prolong visit time interval for sending a mobile sink to hostile environments (or unattended areas), something which is highly risky (or costly). Hence, to develop secure, long-lived MWSNs, we need to answer the following questions:

- How to diminish power consumption of security mechanism as less as possible.
- How to compress sensed data in a more efficient way?
- How to design power management scheme to prolong both battery lifetime and network lifetime?

B. SKC-based Distributed Data Access Control

As discussed in Subsection V-D, existing literature addressing distributed data access control policy of MWSNs is based on PKC which is considered too expensive to implement in commodity sensors. The authors in [25] have proposed an SKC based distributed data storage and retrieval scheme, where access control policy is provided by sharing the symmetric key with authorized users based on perturbed polynomials [57]. It has, however, not been proven to be secure in [58]. Therefore how to design an SKC-based distributed data access control scheme in MWSNs remains another open question.

C. Protecting the Mobile Sink

As mentioned earlier, currently there is no literature that addresses location privacy of mobile sink in MWSNs. When a mobile sink visits the static sensors to collect accumulated data, if the location of a mobile sink is detected by an adversary, the adversary can easily capture or destroy the mobile sink. Since a mobile sink usually holds the network master key and has higher privilege than sensors, once a mobile sink is compromised, secret keys and privileges granted to it will be abused. The open questions in this respect are:

- How to protect the location privacy of a mobile sink?
- How to avoid impersonating the behavior of a real mobile sink in the event that it is captured (or comprised) by an adversary?
- How to design the optimal travel route to collect accumulated sensor data with least security risk to the mobile sink?

VII. Conclusion

Different from conventional WSNs, MWSNs are characteristic of dynamic topologies for both sensors and mobile sinks and exhibit new vulnerability when security is concerned. In this article, we present a comprehensive survey on security challenges, threat models and existing security mechanisms in MWSNs. Through in-depth studies on various aspects of security issues in MWSNs, from requirements to solutions, we have not only given a general picture on this emerging research area but also outlined the state-of-the-art security mechanisms proposed so far in MWSNs. Finally, to trigger further interests in the research community towards this direction, a few open questions are pointed out.

Acknowledgments

The research leading to these results has received funding from the EU FP7-PEOPLE-IRSES program under grant agreement no. 247083, project acronym S2EuNet. This work was partially done while Yi Ren was visiting the School of Information Sciences, University of Pittsburgh. This work is also supported in part by the National Natural Science Foundation of China (NSFC), contract/grant number: 60872007 and National 863 High Technology Program of China, contract/grant number: 2009AA01Z239, and by the Ministry of Science and Technology (MOST), China, International Science and Technology Collaboration Program, contract/grant number: 0903.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks (Elsevier)*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.

[3] V. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," *Wirel. Commun. Mob. Comput. (Wiley)*, vol. 8, no. 1, pp. 1–24, Jan. 2008.

[4] S. Camtepe and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey," *Rensselaer Polytechnic Institute (RPI), Tech. Rep. TR-05-07*, pp. 1–27, 2005.

[5] J. Luo, D. Wang, and Q. Zhang, "Double mobility: coverage of the sea surface with mobile sensor networks," in *Proc. IEEE INFOCOM '09*, Rio de Janeiro, Brazil, Apr. 2009, pp. 118 –126.

[6] L. Vieira, U. Lee, and M. Gerla, "Phero-trail: a bio-inspired location service for mobile underwater sensor networks," *IEEE J. Sel. Areas Commun. (JSAC)*, vol. 28, no. 4, pp. 553 –563, May 2010.

[7] M. Chen, S. Gonzalez, V. Leung, Q. Zhang, and M. Li, "A 2G-RFID-based e-healthcare system," *IEEE Wireless Communications Magazine*, vol. 17, no. 1, pp. 37–43, Feb. 2010.

[8] B. Pásztor, L. Mottola, C. Mascolo, and G. P. Picco, "Selective code dissemination in mobile wireless sensor networks," in *Proc. ACM/IFIP/USENIX Middleware '08*, Leuven, Belgium, Dec. 2008, pp. 113–115.

[9] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme, "Robomote: enabling mobility in sensor networks," in *Proc. 4th IEEE Int. Symp. on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, CA, USA, Apr. 2005, pp. 404 – 409.

[10] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *Proc. ACM MobiHoc '05*, Urbana-Champaign, IL, USA, May 2005, pp. 300–308.

[11] J. Lopez, R. Roman, and C. Alcaraz, "Analysis of security threats, requirements, technologies and standards in wireless sensor networks," *Foundations of Security Analysis and Design V*, vol. 5705, pp. 289–338, Aug. 2009.

[12] J. Jaffe and C. Schurgers, "Sensor networks of freely drifting autonomous underwater explorers," in *Proc. 1st ACM Int. Workshop on Underwater Networks (WUWNet '06)*, Los Angeles, CA, USA, Sep. 2006, pp. 93–96.

[13] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Commun. Surveys Tutorial*, vol. 10, no. 3, pp. 6–28, 2008.

[14] R. Di Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Catch me (if you can): data survival in unattended sensor networks," in *Proc. IEEE PerCom '08*, Hong Kong, Mar. 2008, pp. 185–194.

[15] D. Ma, C. Soriente, and G. Tsudik, "New adversary and new threats: security in unattended sensor networks," *IEEE Network*, vol. 23, no. 2, pp. 43 –48, 2009.

[16] J. Deng, C. Hartung, R. Han, and S. Mishra, "A practical study of transitory master key establishment for wireless sensor networks," in *Proc. IEEE SecureComm '05*, Athens, Greece, Sep. 2005, pp. 289 – 302.

[17] W. Stallings, L. Brown, M. Bauer, and M. Howard, *Computer security: principles and practice*. Pearson Prentice Hall, 2008.

[18] D. Ma and G. Tsudik, "DISH: Distributed Self-Healing," in *Proc. 10th Int. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS '08)*, Detroit, MI, USA, Nov. 2008, pp. 47–62.

[19] R. Di Pietro, D. Ma, C. Soriente, and G. Tsudik, "POSH: Proactive co-Operative Self-Healing in unattended wireless sensor networks," in *Proc. IEEE Symp. on Reliable Distributed Systems (SRDS '08)*, Napoli, Italy, Oct. 2008, pp. 185–194.

[20] Y. Ren, V. Oleshchuk, and F. Y. Li, "A scheme for secure and reliable distributed data storage in unattended WSNs," in *Proc. IEEE GLOBECOM '10*, Miami, FL, USA, Dec. 2010, pp. 1–6.

[21] R. Di Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Data security in unattended wireless sensor networks," *IEEE Trans. on Computers*, vol. 58, no. 11, pp. 1500–1511, 2009.

[22] R. Di Pietro, L. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Playing hide-and-seek with a focused mobile adversary in unattended wireless sensor networks," *Ad Hoc Networks (Elsevier)*, vol. 7, no. 8, pp. 1463–1475, 2009.

[23] D. Ma and G. Tsudik, "Extended abstract: forward-secure sequential aggregate authentication," in *Proc. IEEE Symp. on Security and Privacy (S&P '07)*, Oakland, CA, USA, May. 2007, pp. 86–91.

[24] A. Yavuz and P. Ning, "Hash-based sequential aggregate and forward secure signature for unattended wireless sensor networks," in *Proc. 6th Int. Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '09)*, Toronto, Canada, Jul. 2009, pp. 1 –10.

[25] N. Subramanian, C. Yang, and W. Zhang, "Securing distributed data storage and retrieval in sensor networks," in *Proc. IEEE PerCom '07*, vol. 3, no. 6, NYC, USA, Mar. 2007, pp. 659 – 676.

[26] S. Yu, K. Ren, and W. Lou, "FDAC: toward fine-grained distributed data access control in wireless sensor networks," in *Proc. IEEE INFOCOM '09*, Rio de Janeiro, Brazil, Apr. 2009, pp. 963–971.

[27] R. Zhang, Y. Zhang, and K. Ren, "DP^2AC: Distributed Privacy-Preserving Access Control in sensor networks," in *Proc. IEEE INFOCOM '09*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1251–1259.

[28] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proc. 25th IEEE Int. Conf. on Distributed Computing Systems (ICDCS '05)*, Columbus, OH, USA, Jun. 2005, pp. 599–608.

[29] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Proc. IEEE INFOCOM '08*, Phoenix, AZ, USA, Apr. 2008, pp. 51 –55.

[30] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proc. IEEE Int. Conf. on Network Protocols (ICNP '07)*, Beijing, China, Oct. 2007, pp. 314 –323.

[31] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "Protecting receiver-location privacy in wireless sensor networks," in *Proc. IEEE INFOCOM '07*, Anchorage, AL, USA, May 2007, pp. 1955–1963.

[32] J. Deng, R. Han, and S. Mishra, "Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks," *Pervasive and Mobile Computing (Elsevier)*, vol. 2, no. 2, pp. 159 – 186, 2006.

[33] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks," in *Proc. ACM MobiHoc '05*, Urbana-Champaign, IL, USA, May 2005, pp. 378–389.

[34] H. Song, S. Zhu, W. Zhang, and G. Cao, "Least privilege and privilege deprivation: toward tolerating mobile sink compromises in wireless sensor networks," *ACM Trans. Sen. Netw. (TOSN)*, vol. 4, no. 4, pp. 1–34, 2008.

[35] M. Conti, R. Di Pietro, L. Mancini, and A. Mei, "Emergent properties: detection of the node-capture attack in mobile wireless sensor networks," in *Proc. 1st ACM Conf. on Wireless Network Security (WiSec '08)*, Alexandria, VA, USA, Mar. 2008, pp. 214–219.

[36] M. Conti, R. Di Pietro, A. Gabrielli, L. V. Mancini, and A. Mei, "The smallville effect: social ties make mobile networks more secure against node capture attack," in *Proc. 8th ACM Int. Symp. on Mobility Management and Wireless Access (MobiWac'10)*, Bodrum, Turkey, Oct. 2010, pp. 99–106.

[37] R. Di Pietro, G. Oligeri, C. Soriente, and G. Tsudik, "Intrusion-resilience in mobile unattended WSNs," in *Proc. IEEE INFOCOM '10*, San Diego, CA, USA, Mar. 2010, pp. 1 –9.

[38] ——, "Securing mobile unattended WSNs against a mobile adversary," in *Proc. 29th IEEE Int. Symp. on Reliable Distributed Systems (SRDS'10)*, New Delhi, India, Oct. 2010, pp. 11–20.

[39] M. Bellare and S. Miner, "A forward-secure digital signature scheme," in *Proc. Advances in Cryptology - CRYPTO '99*, vol. 1666, 1999, pp. 786–786.

[40] G. Itkis and L. Reyzin, "Forward-secure signatures with optimal signing and verifying," in *Proc. Advances in Cryptology - CRYPTO '01*, 2001, pp. 332–354.

[41] ——, "SiBIR: Signer-base intrusion-resilient signatures," in *Proc. Advances in Cryptology - CRYPTO '02*, 2002, pp. 499–514.

[42] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *Proc. Advances in Cryptology - EUROCRYPT '02*, 2002, pp. 65–82.

[43] Y. Dodis, M. Franklin, J. Katz, A. Miyaji, and M. Yung, "Intrusion-resilient public-key encryption," in *Proc. Topics in Cryptology - CT-RSA '03*, 2003, pp. 19–32.

[44] ——, "A generic construction for intrusion-resilient public-key encryption," in *Proc. Topics in Cryptology - CT-RSA '04*, 2004, pp. 1997–1997.

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

[46] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Proc. 10th Annual Network and Distributed System Security Symp. (NDSS '03)*, vol. 276, San Diego, CA, USA, Feb. 2003.

[47] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected

false data in sensor networks," in *Proc. IEEE Symp. on Security and Privacy (S&P '04)*, Oakland, CA, USA, May. 2004, pp. 259–271.
[48] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Tran. Sen. Netw. (TOSN)*, vol. 2, no. 4, pp. 500–528, 2006.
[49] ——, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *Proc. 10th ACM Conf. on Computer and Communications Security (CCS '03)*, Washington D.C., USA, Oct. 2003, pp. 62–72.
[50] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptology*, vol. 17, no. 4, pp. 297–319, Aug. 2004, Extended abstract in *Proc. ASIACRYPT '01*.
[51] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. Advances in Cryptology - EUROCRYPT '03*, 2003, pp. 641–641.
[52] A. Lysyanskaya, S. Micali, L. Reyzin, and H. Shacham, "Sequential aggregate signatures from trapdoor permutations," in *Proc. Advances in Cryptology - EUROCRYPT '04*, 2004, pp. 74–90.
[53] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters, "Sequential aggregate signatures and multisignatures without random oracles," in *Proc. Advances in Cryptology - EUROCRYPT '06*, 2006, pp. 465–485.
[54] R. L. Rivest, A. Shamir, and D. A. Wagner, "Time-lock puzzles and timed-release crypto," Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, Tech. Rep., 1996.
[55] R. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 40 –48, 1994.
[56] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, Aug. 1996.
[57] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proc. Advances in Cryptology - CRYPTO '92*, vol. 740, 1992, pp. 471–486.
[58] M. Albrecht, C. Gentry, S. Halevi, and J. Katz, "Attacking cryptographic schemes based on "perturbation polynomials"," in *Proc. 16th ACM Conf. on Computer and Communications Security (CCS '09)*, Chicago, IL, USA, Nov. 2009, pp. 1–10.
[59] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. on Comp. Comm. Sec. (CCS '06)*, Alexandria, VA, USA, Oct. 2006, pp. 89–98.
[60] ORION. [Online]. Available: http://www.joiscience.org/ocean_observing/advisors
[61] NOPP. [Online]. Available: http://www.nopp.org
[62] IOOS. [Online]. Available: http://www.ocean.us
[63] B. Carbunar, Y. Yu, W. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," *ACM Trans. Sen. Netw. (TOSN)*, vol. 6, no. 2, pp. 1–34, 2010, A preliminary version of this paper appeared in *Proc. IEEE SECON '07*.
[64] D. Chaum, "Blind signatures for untraceable payments," in *Proc. Advances in Cryptology - CRYPTO '82*, vol. 82, 1982, pp. 199–203.
[65] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom '00*, Boston, MA, USA, Aug. 2000, pp. 243–254.
[66] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1902–1915, Dec. 2009, A preliminary version of this paper appeared in *Proc. ACM MobiCom '06*.
[67] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Proc. 1st IEEE Int. Conf. on Security and Privacy for Emerging Areas in Communications Networks (SecureComm '05)*, Athens, Greece, Sep. 2005, pp. 113 – 126.
[68] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proc. 1st ACM Conf. on Wireless Network Security (WiSec '08)*, Alexandria, VA, USA, Mar. 2008, pp. 77–88.
[69] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *Proc. 3rd IEEE Int. Symp. on Information Processing in Sensor Networks (IPSN '04)*, Berkeley, CA, USA, Apr. 2004, pp. 259–268.

Yi Ren received his M.Sc. degree in Automation from Wuhan University of Technology (WUT), Wuhan, China. He is currently a Ph.D. candidate at the Department of Information and Communication Technology, University of Agder (UiA), Norway. Recently he is working at University of Pittsburgh (PITT) as a guest researcher. His current research interests include wireless sensor network security, and RFID security. He is a student member of the IEEE.

Vladimir A. Oleshchuk is Professor of Computer Science at University of Agder, Norway. He received his M.Sc. in Applied Mathematics (1981) and Ph.D. in Computer Science (1988) from the Taras Shevchenko Kiev State University, Kiev, Ukraine, and his M.Sc. in Innovations and Entrepreneurship (2007) from the Norwegian University of Science and Technology (NTNU). From 1987 to 1991 he was Assistant Professor and then Associate Professor at the Taras Shevchenko Kiev State University. He has been working at University of Agder since 1992. He is a member of the IEEE and a senior member of the ACM. His current research interests include formal methods and information security, privacy and trust with special focus on telecommunication systems.

Frank Y. Li holds a Ph.D. degree from the Norwegian University of Science and Technology (NTNU). He worked as a senior researcher at UniK, University Graduate Center, University of Oslo before joining the Department of Information and Communication Technology, University of Agder as an Associate Professor in August 2007. He is a senior member of the IEEE. His research interest includes 3G and beyond mobile systems and wireless networks, mesh and ad hoc networks; QoS, resource management and traffic engineering in wired and wireless IP-based networks; analysis, simulation and performance evaluation of communication protocols and networks.

Xiaohu Ge is currently an Associate Professor with the Department of Electronics and Information Engineering at Huazhong University of Science and Technology (HUST), China. He received his Ph.D. degree in Communication and Information Engineering from HUST in 2003. He has worked at HUST from Dec. 2005. Prior to that, he worked as an assistant researcher at Ajou University (Korea) and Politecnico Di Torino (Italy) from Jan. 2004 to Dec. 2005. His research interests are in the area of mobile communication, traffic modeling in wireless networks, as well as interference modeling in wireless communications. Dr. Ge serves as an Editor for the international journal KSII Transactions on Internet and Information Systems. He is a Member of the IEEE.