RSDA: Reputation-based Secure Data Aggregation in Wireless Sensor Networks

Hani Alzaid Ernest Foo Juan Gonzalez Nieto
Information Security Institute
Queensland University of Technology
PO Box 2434, Brisbane, Queensland 4001
halzaid@isi.qut.edu.au {e.foo, j.gonzaleznieto}@qut.edu.au

Abstract

Wireless Sensor Networks (WSNs) are a new technology that is expected to be used in the near future due to its cheap cost and data processing ability. However, securing WSNs with traditional cryptographic mechanism is insufficient because of the existing limited resources and the lack of tamper resistant hardware. In this paper, we propose a Reputation-based Secure Data Aggregation for WSNs (RSDA) that integrates aggregation functionalities with the advantages provided by a reputation system to enhance the network lifetime and the accuracy of the aggregated data. We bind symmetric secret keys to geographic locations and assign these keys to sensor nodes based on their locations. RSDA therefore can resist an adversary that is capable to compromise up to $\mathcal{W}$ sensor nodes in total with no more than $t-1$ compromised nodes in any cell.

1. Introduction

Securing network communications has traditionally been achieved through cryptographic mechanisms, but these are insufficient to protect wireless sensor networks (WSNs). Because sensor nodes are deployed for long periods in hostile environments it is possible for an adversary to physically take over a sensor node and obtain access to the cryptographic keys. Adversaries are given the opportunity to pretend to be a legitimate node while performing malicious actions. Another reason is the limited resources that WSN suffers from such as low computational capability, limited energy source and small memory. The wireless security community has therefore developed a suite of mechanisms to complement cryptographic techniques such as a reputation system that can be defined as a system that collects, processes, and disseminates feedback about the history of sensors' behavior.

Most current research in reputation systems focus on ad hoc networks not WSNs. Unfortunately, existing reputation schemes, that are proposed for WSNs, are not appropriate to be used in the data aggregation context. For example, studies such as [2] examined how good these nodes are in performing routing functionalities. They are not aware about the content of the sensed data. The disadvantage of this is that some sensors may still get good reputation values although they provide invalid readings since no check is made on the sensed data. On the other hand, most of the existing secure data aggregation schemes focus either on the aggregator behavior such as Secure Information Aggregation in sensor network (SIA) [5] or on the security of the aggregation functions such as Resilient Aggregation in WSN [7]. They assume that leaf nodes are honest and report correct readings.

In this paper, we propose a Reputation-based Secure Data Aggregation for wireless sensor networks (RSDA) that integrates the aggregation functionalities with the advantages that are provided by a reputation system in order to enhance the network life time and the accuracy of the aggregated data without trimming the abnormal (but correct) readings as suggested by Wagner [7]. Eliminating abnormal readings with no further investigation is impractical especially in applications such as monitoring bush fire or monitoring temperatures within oil refineries. The target terrain, where RSDA is implemented, is divided into smaller cells of equal size. Each cell has $\mathcal{T}$ nodes where only one of them is selected, based on its reputation value, to be the Cell Representative. Each node has a monitoring mechanism similar to the Watchdog that was proposed by Martie et al. [4] in order to compare its result with the result of its neighbors. Each node in a cell performs redundant operations to monitor the cell representative operations. RSDA follows a request-response paradigm where the base station initiates the aggregation process by flooding a query message into the network. The transformation from this paradigm to the periodic paradigm however is straight forward by letting the representatives periodically report their data without the need to wait for the base station's query. An example of the

applications where RSDA can fit is checking the temperature in oil refineries where normal and abnormal readings are equally important for the safety department.
The rest of the paper is organized as follows. Section 2 lists the requirements for RSDA. Section 3 discusses the expected type of adversary that RSDA resists. Section 4 describes RSDA. In Section 5, the performance of RSDA is discussed. Section 6 concludes the paper.

2 Requirements

RSDA focuses on providing two main properties which are ***data accuracy*** and ***data availability***. Other requirements are necessary in order to achieve these two properties as will be discussed in Section 5. These requirements are:
Data Integrity that ensures the content of a message has not been altered, either maliciously or accidentally, during transmission which helps RSDA to filter out incorrect data and save the processing energy if this data traveled all the way to the base station ($\mathcal{BS}$).
Data Freshness that ensures the data are recent and have not been replayed. Injecting old data into the network requires nodes to process this unnecessary data which leads to more energy consumption. This old data also did not represent the current (correct) cell reading which affects the accuracy of the aggregated data.
Entity Authentication that allows the receiver to verify whether the message is sent by the claimed sender or not. Therefore, an adversary will not be able to participate and inject data into the network and then affect the data accuracy unless it has valid keys.

3 Adversarial Model

Given that the number of sensor nodes in each cell is $\mathcal{T}$, we assume that the adversary ($\mathcal{ADV}$) is capable of compromising $\mathcal{W}$ sensor nodes where $\mathcal{W} \ggg \mathcal{T}$ but there is no more than $t-1$ compromised nodes in any cell. When the $\mathcal{ADV}$ compromises a sensor node x, it is able to read all x's internal memory and then the $\mathcal{ADV}$ could manipulate x to alter the received content or even drop it. Moreover, $\mathcal{ADV}$ can degrade the reputation system accuracy by lying about the reputation values. For example, it can falsely accuse well-behaving nodes by assigning them negative reputation values and then distributing them to its neighbors. The $\mathcal{ADV}$ however can not compromise the $\mathcal{BS}$ which is secured and under the supervision of a network administrator.

4 The Proposed Scheme - RSDA

We assume that sensor nodes lack tamper-resistance property, have unique id, and are preloaded with two network wide shared keys K^1 and K^2. We also assume a large deployment area, the dimensions of which are known in advance and that nodes are uniformly distributed over this area. A grid structure is used to divide the target terrain into smaller non-overlapping cells of equal areas. The dimension of each cell is small enough to allow the radio range of each sensor to cover its surrounding cells. We assume the existence of a short period of time when the network is not vulnerable to any attacks. During this time, each sensor node discovers its neighboring nodes, finds out which cell it belongs to, and computes its cell key and shared keys with neighboring cells as described below.
RSDA focuses on the multiple aggregator model that was identified by Alzaid et al. [1] where the aggregation is performed at each cell. Each node monitors the behavior of other nodes within the same cell and then calculates the reputation value for them based on participation in some cell operations such as sensing, forwarding, and aggregation. RSDA is composed of two types of identities: a $\mathcal{BS}$ and normal sensor nodes. The $\mathcal{BS}$ is entrusted with the task of initiating queries to the network, processing received answers for these queries, and deriving meaningful information that reflects the events in the target field. The normal sensors are grouped into cells and in each cell one of sensors is selected to be the cell representative C^{rep}.
Initially C^{rep} is chosen randomly since all nodes start with same reputation value. These cells can be an intermediate cell, that receives data from downstream cells and performs sensing, aggregation, and forwarding operations, or a leaf

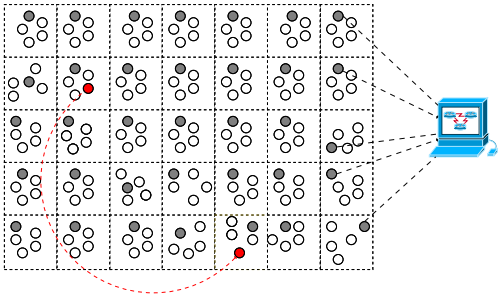

Figure 1: RSDA

Table 1: Reputation Table Format

Node	R_S		R_F		R_A	
ID	α	β	α	β	α	β
x_1	10	4	8	6	-	-
x_2	13	1	14	-	14	-
x_3	8	6	10	4	-	-
.	.	.	.	.	.	.
x_i	.	.	.	.	.	.

cell that does not receive data from downstream cells and does not perform aggregation functions (see Figure 1). The C^{rep} is responsible for confirming its cell reading C^{read} (reported by other cell member), aggregating it with other readings (if the cell is an intermediate cell), and forwarding it to upper stream cell. In addition to reporting C^{read} to C^{rep}, other nodes evaluate the behavior of their C^{rep} and other nodes in the same cell. The node behavior is represented in the form (α,β) tuple where α and β represent the amount of positive and negative ratings that are calculated by each node for other nodes in its cell and then stored in the reputation table (see Table 1). If x has behaved well for a specific function, α for node x is incremented by one. Otherwise β is incremented. The nodes behavior are examined for three functions: data sensing, data forwarding, and data aggegation (if x is the C^{rep} for an intermediate cell). Each node therefore maintains a reputation table for its cell members and keeps recording α and β for these functions (sensing, forwarding, aggregation) separately as in Table 1. To fill this reputation table, each sensor node evaluates the sensing, forwarding, and aggregation (if it is in an intermediate cell) functionalities of each cell members and computes the amount of positive and negative (α,β) ratings for each function. RSDA uses beta probability density function (PDF) to update the reputation value of each sensor node due to its flexibility, strong foundation on the statistics, simplicity that meet the needs of resource constraint nodes [3]. The reputation value can be expressed as follows:

$$R_{S/A/F} = \frac{\alpha_{S/A/F}}{\alpha_{S/A/F} + \beta_{S/A/F}}$$

when nothing is known, the *a priori* distribution is the uniform beta PDF with $\alpha = 1$ and $\beta = 1$. Due to space limitation, we refer the interested readers to Ismail and Jøang [3] for details about beta distribution. The notation to be used in the rest of the paper can be found in Table 2.

Bootstrap Phase. This phase constitutes a short duration of time immediately following the network deployment. It is short enough to assume that no attacks are possible during this phase. For example, let us consider an arbitrary sensor node x within C_i. The node x computes K_{C_i} which is used to authenticate any communication between itself and any node in the same cell in a similar way to Ren et al. [6] as follows:

$$K_{C_i} = H(K^1 \,||\, C_i)$$

where $||$ represents bit string concatenation. The K_{C_i} is used to prevent non cell members from participating in the cell operations and then affecting the accuracy of C_i^{read}. After that, each sensor node computes the authentication key with adjacent cells C_j as follows:

$$K_{C_i}^{C_j} = H(K^2 \,||\, C_i \,||\, C_j)$$

At the end of this phase, each sensor node deletes K^1 and K^2 to prevent $\mathcal{ADV}$ from getting access.

Table 2: Notations for RSDA

Name	Description
K^1, K^2	two network wide shared keys.
C_i	The i^{th} cell.
K_{C_i}	Cell key for the i^{th} cell.
$K_{C_i}^{C_j}$	Shared key between i^{th} and j^{th} cell.
$H(.)$	Hash function.
$MAC_{K_{c_i}}$	Message authentication code computed by using K_{C_i}.
$MAC_{K_{c_i}^{c_j}}$	Message authentication code computed by using $K_{C_i}^{C_j}$.
$\mathcal{ADV}$	An adversary around the WSN.
$\mathcal{T}$	Number of nodes in each cell.
$\mathcal{N}$	Total number of nodes in RSDA.
$\mathcal{W}$	Total number of compromised nodes.
t	The minimum number of nodes in a cell that is required to revoke misbehaving C^{rep} or to confirm a new C^{read}
x, y	sensor node x and y respectively.
r_x, r_y	readings from x and y respectively.
$\mathcal{BS}$	The base station.
C_i^{read}	Reported (sensed) data from C_i.
$\mathcal{F}$	An aggregation function.
$\mathcal{AD}_{C_i}$	Aggregated data obtained by $\mathcal{F}$ at C_i.
Q_n	A query number.
$R_{S/A/F}^x$	Reputation value for a *sensing/ aggregation/ forwarding* functionality of x.
$\alpha_{S/A/F}^x$	The amount of positive ratings for a *sensing/ aggregation/ forwarding* functionality of x.
$\beta_{S/A/F}^x$	The amount of negative ratings for a *sensing/ aggregation/ forwarding* functionality of x.
$Thr_{A/S}$	The pre-defined threshold for the *aggregation/ sensing* functionality.

Data Aggregation. Before describing how aggregation works, we first introduce the packet format used within the network. Each packet has the following format:

$$\left\{ C_i^{rep} \,,\, C_j^{rep} \,,\, Q_n \,,\, Payload \right\}$$

where C_i^{rep} represents the sending cell representative, C_j^{rep} represents the receiving cell representative, and $Payload$ denotes the packet content. An aggregation process begins

when the $\mathcal{BS}$ propagates a query to all cells as follows:

$$\{\ \mathcal{BS},\ all\_cells,\ Q_n,\ Payload\ \}$$

The propagated query and the answer to this query are relayed to its destination via some intermediate cells. The data flow relies further on the routing algorithm that is not discussed here due to the space limits. Functions, that are done at each cell to answer this query, vary depending on whether the cell is an intermediate cell or a leaf cell.

At leaf cell When a leaf cell C_i receives the query, C_i^{rep} randomly selects a sensor node x from its cell to send back r_x as follows:

$$\{\ C_i^{rep},\ x,\ Q_n, Payload\ \}$$

As a response, x senses some physical phenomena (as requested) and then sends it back to C_i^{rep} as follows:

$$\{x,\ C_i^{rep},\ Q_n, Payload\}\ \text{, where}$$

$$Payload\ \equiv\ r_x\ ||\ MAC_{K_{c_i}}(x\ ||Q_n\ ||r_x)$$

Other nodes in C_i are listening to the on-going traffic between the elected node and C_i^{rep} since they share the same cell key. A neighboring node y does not agree on the reading r_x if $|r_y - r_x| < Thr_S$. If they agree on r_x, they update α_S^x and α_F^x of node x and consider r_x as the C_i^{read}. They also update α_S for all other nodes because their agreements on the C_i^{read} means that their readings were within Thr_S. If they do not agree, they:

- Update β_S^x (if the reading was unacceptable) or β_F^x (if the destination was not the cell representative or no reply was sent).
- Collaborate with other nodes in the cell to supply C_i^{rep} with the correct C_i^{read}.

Each disagreeing node sends its reading to the C_i^{rep} and thus other nodes are updating α_S^y and β_S^y. After receiving claims from $n\ \geq\ t$ eligible nodes[1] regarding the reported reading, C_i^{rep} computes the cell reading by using *Exogenous Discounting of Unfair Ratings* where the reputation values of these n nodes are used to determine the weight given to the information as proposed by Whitby et al. [8] as follows:

$$C_i^{read} = \frac{\sum_{i=1}^{n}(r_i\ *\ R_S^i)}{\sum_{i=1}^{n}(R_S^i)}$$

It is based on the assumption that sensors with low reputation are likely to give unfair information and vice versa. Then, the C_i^{rep} forwards this reading to next cell in the upstream path as follows:

$$\{\ C_i^{rep}\ ,\ C_j^{rep}\ ,\ Q_n\ ,\ Payload\}\ \text{, where}$$

[1]To be an eligible node, a node x should has $R^x > Thr_R$.

$$Payload\ \equiv\ C_i^{read}\ ||\ MAC_{K_{C_i}^{C_j}}(C_i^{rep}\ ||Q_n\ ||C_i^{read})$$

Other nodes in the cell monitor this transmission to evaluate the behavior of the C_i^{rep} since they also know the shared key between C_i and neighboring cells. If the cell reading is different by more than Thr_S, then $\beta_S^{C_i^{rep}}$ is updated. Otherwise, $\alpha_S^{C_i^{rep}}$ is updated. Whenever C_i^{rep} forwards the aggregated data to the right destination, $\alpha_F^{C_i^{rep}}$ is updated.

At an intermediate cell In order to ensure that the message is received from the claimed entity (data-origin authentication), the C_j^{rep} computes the MAC for the received data from the downstream cell, and compares it with the attached one. If they do not match, then the reading coming from C_i^{rep} is ignored. Otherwise, C_j^{rep} removes the attached MAC and considers the reported data as an input to the aggregation function. Then, C_j^{rep} waits until receiving readings from its cell (which is done in the same way as the leaf cell does) and other children cells in order to apply the aggregation function on them as follows:

$$\mathcal{AD}_{C_j} = \mathcal{F}(C_1^{read}, C_2^{read}, ..., C_j^{read})$$

After that, C_j^{rep} forwards $\mathcal{AD}_{C_j}$ to upper cell representative C_k^{rep} with the following packet format:

$$\{\ C_j^{rep}\ ,\ C_k^{rep}\ ,\ Q_n\ ,\ Payload\}\ \text{, where}$$

$$Payload\ \equiv\ \mathcal{AD}_{C_j}\ ||\ MAC_{K_{C_j}^{C_k}}(C_j^{rep}\ ||Q_n\ ||\mathcal{AD}_{C_j})$$

Other nodes in cell C_j are still able to keep an eye on the aggregation and forwarding behavior of C_j^{rep}. They calculate the aggregation function $\mathcal{AD}_{C_j}^*$ and match the result with $\mathcal{AD}_{C_j}$. If they are bounded by small value such as $|\mathcal{AD}_{C_j} - \mathcal{AD}_{C_j}^*| < Thr_S$, $\alpha_A^{C_j^{rep}}$ is increased by one. Otherwise, $\beta_A^{C_j^{rep}}$ is increased by one. Also, the $\alpha_F^{C_j^{rep}}$ is increased by one if C_j^{rep} forwards the packet to right C^{rep} that is not in the black list and is one-cell closer to the base station. RSDA uses a reactive form of dissemination frequency where reputation values are computed and propagated after the occurrence of an event. There are two types of reputation-related information that need to be stored in each node:

- **Black-List** which contains a list of nodes that misbehaved during their act as a C^{rep}. Once R_A falls below Thr_A, a new C^{rep} should be elected, and black list the previous C^{rep}. The black list is shared with neighboring cells in order to be informed about misbehaving nodes.
- **Reputation-Table** which contains a list of the cell members and their reputation values as in Table 1.

Cell Representative Replacement As soon as the $R_A^{C_i^{rep}}$ fall below Thr_A, the revocation mechanism is called. The main aim of this mechanism is to: inform neighboring cells about misbehaving C^{rep}, select a new C^{rep} that has higher R_S and R_F, and black list misbehaving C^{rep}. The revocation process starts when n nodes ($n \geq t$) in C_i send revoke messages to neighboring C_j^{rep} in order to inform them about misbehaving C_i^{rep}. Each sensor node, say x, selects one node y that has the highest R_S^y and R_F^y and never been in the black list, as a good candidate to be the new C_i^{rep} and sends the revoke message as follows:

$$\{\, x\ ,\ C_j^{rep}\ ,\ Q_n\ ,\ Payload\} \ \text{, where}$$

$$Payload \ \equiv\ C_i^{rep} \,||\, R_A^{C_i^{rep}} \,||\, y \,||$$

$$MAC_{K_{C_i}^{C_j}}(x \,||\, Q_n \,||\, C_i^{rep} \,||R_A^{C_i^{rep}} \,||y)$$

Each neighboring cell representative, say C_j^{rep}, should receive at least t valid[2]requests to participate in the replacement process. The β_F will be updated for those nodes that did not participate in reporting revocation message. After receiving these n messages, the new C_i^{rep} is selected by applying simple majority vote on them. The replacement process requires exchanging a number of messages which can affect the network lifetime. This process however never starts unless abnormal behavior detected.

5 Discussion

The discussed adversarial capability in Section 3 fits into *the medium adversary class* that was identified by Alzaid et al. [1] since $\mathcal{ADV}$ in RDSA has limited computational power and only can compromise up to $\mathcal{W}$ nodes with no more than $t-1$ nodes in each cell. The comparison between existing secure data aggregation schemes with same adversary type, that was done in [1], is adapted to evaluate RSDA.

Attack Resistance: We study the resilience behavior of RSDA against different attacks that affect either the data accuracy or data availability property. Due to space limitation, we refer the interested readers to Alzaid et al. [1] for details about these attacks.

Each node in RSDA is equipped with the Watchdog mechanism to monitor the behavior of neighboring nodes which helps RSDA to resist against attacks listed in Table 3. When a compromised C_i^{rep}, for example, stops forwarding some selective packets to the upper node (called a selective forwarding attack (SF)), other nodes within the cell are evaluating its behavior and subsequently a negative feedback is assigned. If this misbehavior went undetected, the accuracy of the aggregated data will be affected since some information is not considered in the aggregation results.

[2]A valid request means a request that is received from a sensor node that has an acceptable reputation value and is located in the same cell where the revoked C^{rep} is

Table 3: Attacks Against Existing Aggregation Schemes.

Scheme	NC	SF	R	S
SDA	•	•		•
WDA	•	•	•	•
SecureDAV	•	•	•	•
SDAP	•	•		•
ESA	•	•		•
RSDA	•			

Moreover, replayed data can be processed if the entity authentication requirement is met. If this data is not prevented, the data accuracy and data availability properties will be affected. The data accuracy will be affected in the sense that incorrect data is considered instead of the current data while the data availability is affected because nodes in the path to $\mathcal{BS}$ need to receive, process, and forward this data. Therefore, RSDA introduces a query number in each packet to resists against replay attack (R).

Furthermore, RSDA suffers from the node compromise attack (NC) as do most of the WSN schemes that lack tamper-resistance. However, it is important to know how a single compromised node can affect the robustness of RSDA. An $\mathcal{ADV}$ needs to compromise at least t nodes in any cell in order to be able to fool the C_i^{rep} and let it accept false data, or to revoke current well-behaved C_i^{rep} with one of these t nodes to be C_i^{rep*}. Any cell with t compromised nodes is called a compromised cell.

Moreover, a stealthy attack (S) occurs when $\mathcal{ADV}$ injects false data into the network without revealing its existence which can be launched in RSDA if the $\mathcal{ADV}$ is able to keep the injected false data within $[-Thr_S, Thr_S]$ and is elected by C_i^{rep} to report the C_i^{read*}. We claim that this injected error is acceptable since Thr_S is a network parameter and set by the network administrator. $\mathcal{ADV}$ however can accumulate the accepted error rate for more than one cell in sequence to drift the collected data into outside the accepted range without been detected. This requires to compromise more than one cell in the path to $\mathcal{BS}$. The C_i^{rep} then elects one of these compromised nodes to report the C_i^{read*}. We claim that this is even harder than compromising t nodes in a cell which is the security limits for RSDA because the $\mathcal{ADV}$ needs to compromise more than one cell in each path to $\mathcal{BS}$ to feed RSDA with an error rate bigger than Thr_S.

Services provided: RSDA overcomes existing secure data

Table 4: Secure aggregation schemes comparison

Scheme	C	I	F	AV	AU
SDA		•	•		•
WDA		•			•
SecureDAV	•	•			•
SDAP	•	•	•		•
ESA	•	•	•		•
RSDA		•	•	•	•

aggregation mechanisms by providing data availability (AV). It eliminates misbehaving nodes and prevents them from participating in the network. This reduces the number of untrusted packets traveling within the network which leads to reduce the energy consumption that is resulted from processing these packets and consequently prolong the network lifetime. RSDA also provides other security services such as data integrity (I), freshness (F), and authentication (AU). Data confidentiality (C) can be offered by RSDA since nodes share intra/inter keys with neighboring nodes. However, data confidentiality is not considered in this paper because we only focus on data accuracy and data availability as discussed in Section 2. To the best of our knowledge, RSDA is the only secure data aggregation that provides data availability (AV) beside the minimum security services ((I), (F), and (AU)), that were identified in [1], for such a secure data aggregation (see Table 4).

Communication overhead: We assume, for demonstration purpose, that node and cell ID are 2 bytes, key size is 4 bytes, Q_n is 1 byte, sensed and aggregated data are 2 bytes, MAC is 9 bytes, and a reputation value is 1 byte long. Let us consider the ideal case where there is no NC has been launched yet and the sensed data is within the accepted error rate. C_i^{rep} randomly selects x to report the cell reading with 5 bytes communication overhead. x then senses the required information and reports it back to C_i^{rep} with 16 bytes overhead. C_i^{rep} then aggregates (if C_i is an intermediate cell) and forwards it to the upper stream cell with 16 bytes overhead. If the average path length to $\mathcal{BS}$ is P, then the estimated communication overhead per query is $5 + 16P$ bytes. Let us now consider the case where the NC attack has been launched with $\mathcal{W}$ compromised nodes but no more than $t - 1$ compromised nodes at each cell. Nodes that are disagreed with the reported reading need to send their readings to C_i^{rep} with communication overhead around $16m$ where m represents the number of disagreed nodes. Other communication overhead comes when C_i^{rep} needs to be replaced. It introduces $19n$ bytes where n is the number of nodes that sent revoke messages and $n \geq t$. The total number of communication overhead therefore can be calculated as $5 + 16P + D * 16m + D' * 19n$, where D represents the number of cells where disagreement about the reported reading occurred and D' represents the number of cells where their C^{rep} need to be replaced.

6 Conclusion

In this paper, we have proposed a reputation-based secure data aggregation that focuses on enhancing the data availability and the accuracy of the aggregated data. By monitoring neighborhood's activities, each sensor node evaluates the behavior of its cell members in order to filter out the inconsistent data in the presence of multiple compromised nodes ($< t$ in each cell). RSDA is expected to detect compromised nodes and then black list them which helps to reach the main two goals: extend the network lifetime and protect the accuracy of the aggregated data. In the future work, we are going to evaluate RSDA by using one of the network simulations such as NS2 and consider data confidentiality and study the impact of NC attack on the privacy of the aggregated data.

References

[1] H. Alzaid, E. Foo, and J. M. Gonzalez Nieto. Secure data aggregation in wireless sensor network: a survey. In L. Brankovic and M. Miller, editors, *Sixth Australasian Information Security Conference (AISC 2008)*, volume 81 of *CRPIT*, pages 93–105, Wollongong, NSW, Australia, 2008. ACS.

[2] P. Dewan and P. Dasgupta. Trusting routers and relays in ad hoc networks. In *ICPP Workshops*, pages 351–358. IEEE Computer Society, 2003.

[3] R. Ismail and A. Jøsang. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[4] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MOBICOM*, pages 255–265, 2000.

[5] B. Przydatek, D. X. Song, and A. Perrig. SIA: Secure information aggregation in sensor networks. In I. F. Akyildiz, D. Estrin, D. E. Culler, and M. B. Srivastava, editors, *SenSys*, pages 255–265. ACM, 2003.

[6] K. Ren, W. Lou, and Y. Zhang. LEDS: Providing location-aware end-to-end data security in wireless sensor networks. *IEEE Trans. Mob. Comput.*, 7(5):585–598, 2008.

[7] D. Wagner. Resilient aggregation in sensor networks. In S. Setia and V. Swarup, editors, *Proceedings of the 2nd ACM Workshop on Security of ad hoc and Sensor Networks, SASN 2004*, pages 78–87. ACM, 2004.

[8] A. Whitby, A. Josang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. In *the Workshop on Trust in Agent Societies, at the Third International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS2004)*.