Quarantine Region Scheme to Mitigate Spam Attacks in Wireless Sensor Networks

Vedat Coskun, *Member*, *IEEE*, Erdal Cayirci, *Senior Member*, *IEEE*, Albert Levi, *Member*, *IEEE*, and Serdar Sancak

Abstract—The Quarantine Region Scheme (QRS) is introduced to defend against spam attacks in wireless sensor networks where malicious antinodes frequently generate dummy spam messages to be relayed toward the sink. The aim of the attacker is the exhaustion of the sensor node batteries and the extra delay caused by processing the spam messages. Network-wide message authentication may solve this problem with a cost of cryptographic operations to be performed over all messages. QRS is designed to reduce this cost by applying authentication only whenever and wherever necessary. In QRS, the nodes that detect a nearby spam attack assume themselves to be in a quarantine region. This detection is performed by intermittent authentication checks. Once quarantined, a node continuously applies authentication measures until the spam attack ceases. In the QRS scheme, there is a trade-off between the resilience against spam attacks and the number of authentications. Our experiments show that, in the worst-case scenario that we considered, a not quarantined node catches 80 percent of the spam messages by authenticating only 50 percent of all messages that it processes.

Index Terms—Network-level security and protection, wireless sensor networks, authentication, quarantine region, spam attacks.

1 Introduction

Wireless sensor networks are rapidly deployable, flexible, and self-organizing systems that incur low deployment and maintenance cost and, therefore, they have many application areas, such as military, environmental, health, home, etc. In many of these application areas, security is one of the key challenges. When a sensor network is reachable, sensor nodes can be collected or destroyed by the enemy. The wireless sensor networks that we focus on in this paper are the ones deployed in regions not accessible for the opponent. The examples for such networks are given in [1]: monitoring friendly forces, equipment, and ammunition (MFFEA) and nuclear, biological, and chemical (NBC) attack detection. For example, in MFFEA application, sensor nodes are deployed among the friendly troops in friendly regions and, therefore, they are not physically accessible by the enemy. Moreover, since this network is widely deployed, there is not an effective way to burn or disable them by electro magnetic pulse (EMP) type attacks. However, in this case, some malicious nodes can be deployed inside the sensor network and they can set several attacks against the sensor network. Karlof and Wagner [2] give a detailed list of such attacks, such as selective forwarding, sinkhole, wormhole, and Sybil attacks. Denial of Service (DoS) is another kind of attack, where the basic aim is to restrain the nodes to function properly. DoS attacks on sensor networks are extensively analyzed by Wood and Stankovic in [3].

In this paper, we focus on a kind of a DoS attack, the so-called *spam attack*, which can be carried out by the hostile nodes called antinodes. In this attack, antinodes deployed inside sensor networks frequently generate unsolicited spam messages. These spam messages are relayed by the sensor nodes toward the sink. In this way, antinodes cause the sensor nodes to use up their battery unnecessarily. If such a spam attack continues, sensor nodes, especially the ones close to the sink, fail sooner than their natural lifetime due to energy depletion. In other words, antinodes fight against legitimate sensor nodes in the sensor field, so we can say that this attack is a kind of war between sensor nodes and antinodes.

In this paper, we introduce the quarantine region scheme (QRS) to defend against spamming antinodes. In QRS, continuous security measures are taken temporarily and only in the quarantine regions, which are the regions that confine all the quarantined nodes. Not quarantined nodes perform message authentication intermittently in order to catch possible spam. When a node comes across a spam message, it declares itself as quarantined.

Our scheme reduces the overhead for security comparing to other possible schemes that always take security measures for every node. We performed a mix of analytical and simulation techniques for the performance evaluation of QRS. Once quarantined, a node always eliminates spam. Our analysis shows that, even in the worst case, where the antinode coverage area is almost the whole network, a not quarantined node can catch 80 percent of the spam messages by authenticating only 50 percent of all messages.

- *V. Coskun is with the Department of Information Technologies, ISIK University, Sile 34980, Istanbul, Turkey. E-mail: vedatcoskun@isikun.edu.tr.*
- *E. Cayirci is with the Electrical and Computer Engineering Department, University of Stavanger, N-4036 Stavanger, Norway. E-mail: erdal.cayirci@genetlab.com.*
- *A. Levi is with the Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli, Tuzla 34956 Istanbul, Turkey. E-mail: levi@sabanciuniv.edu.*
- *S. Sancak is with the Naval Science and Engineering Institute, Turkish Naval Academy, Deniz Harp Okulu, Tuzla 34942 Istanbul, Turkey. E-mail: ssancak@dho.edu.tr.*

Manuscript received 30 Dec. 2004; revised 7 May 2005; accepted 10 May 2005; published online June 15 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0356-1204.

The remainder of this paper is organized as follows: In Section 2, we introduce QRS and explain its design details, including formation of quarantine regions, message structure, and authentication mechanisms. A discussion on possible attacks and countermeasures on QRS is also given there. The performance of QRS is analytically evaluated in Section 3, where we analyze two performance metrics for QRS: the resilience of a not quarantined node against attacks and the quarantine gain of a not quarantined node during an attack. In Section 4, we give the numerical results from our experiments. Section 5 discusses related works. Section 6 concludes our paper.

2 Quarantine Region Scheme

Spam attacks aim to neutralize the targeted wireless sensor network by using antinodes scattered randomly inside the network. Antinodes, which are much smaller in number as compared to the number of sensor nodes in the network, set spam attacks by generating frequent unsolicited dummy messages and broadcasting them to the neighboring nodes. Hence, they increase the data traffic conveyed in the network. In sensor networks, all data generated by the sensor nodes are forwarded to a central node, i.e., the sink, which collects the sensed data from sensor nodes and then relays them to the users or external networks [1]. The nodes closer to the sink have to relay more messages than the other nodes. Therefore, nodes closer to the sink are expected to fail earlier than the other sensor nodes in the network due to energy depletion. This causes the sink to be disconnected from the sensor network. If the attack continues, other sensor nodes exhaust their batteries as well.

Antinodes may be fixed or mobile. They can use fixed local identification values or change their identifications as frequently as they need. It is relatively easier to develop a procedure to refuse the messages produced by a fixed antinode than a mobile antinode. For example, after detecting a spam attack, prohibiting the sensor with that identification value from sending any other messages to the network could be an easy solution, but only if we can make sure that the antinodes do not change their identifications. However, we do not think that this will be a common case. Therefore, in this paper, we focus on the counter-measures against mobile antinodes that change their local identifications continuously, which is a more challenging problem.

QRS has been designed for the cases where there is continuous struggle between antinodes and legitimate sensor nodes. Thus, sensor nodes are always alerted in order to sense any possible spam attempt around themselves (but are not always quarantined, consequently, message authentication is performed only whenever necessary). QRS is designed under the assumption that no central node (e.g., sink) is responsible for detecting the existence of a spam attack in the sensor field. Instead, each node detects whether or not there is a spam attack occurring in its neighborhood (i.e., within its transmission range). In this respect, QRS is a distributed scheme, which complies with the general architecture of sensor networks phenomenon.

One may argue that a more centralized approach, where the sink detects a spam attack in the network and broadcasts an alarm message in order to let the nodes start checking local spamming antinodes, would be more efficient since the spam detection by sensor nodes is performed only when they are alarmed by sink. Although sensor nodes seem to spend fewer resources for spam detection in such a centralized approach, there exist some disadvantages as well. These disadvantages are listed below:

- A centralized spam detection mechanism requires the sink to correlate spamming behaviors of individual nodes in the network. Such a correlation imposes the need for network-level time synchronization among the central sink and other nodes. However, in our distributed design, there is no need to have such network-level synchronization among the nodes and the sink since spam detection is always performed locally.
- In order to centrally decide about the existence of a spam attack, several spam messages are to be relayed up to the central sink, thus causing unnecessary resource consumption and delay in the network. However, the proposed distributed design helps to attain quicker response to spam attacks since the spam attacks are to be detected at their source.
- In the centralized approach, the sink should send out broadcast messages to start and cancel spam alarms. These messages may be engineered by intelligent antinodes so that they cease attack just after the alarm message and immediately continue just after the cancellation message. In this way, they create periodic spam pulses in the sensor network. Since no broadcast "alarm start" and "alarm cancel" messages are needed in our distributed approach, such antinode manipulations in the system are eliminated.

In this section, we explain the details of proposed distributed spam detection, defense, and quarantine region formation mechanisms.

2.1 Formation of Quarantine Regions

In QRS, a quarantined set of nodes and quarantine regions are determined dynamically by using a distributed approach. Each sensor node decides whether it should be quarantined or not on its own by intermittently checking authentication failures in its transmission range. The duration of authentication checks is a random variable defined as a system parameter as described in the rest of this section. The complete algorithm of forming a quarantine region is shown in Fig. 1.

Not quarantined nodes have two different modes of operation in two alternating periods: 1) In *check the status periods*, nodes check spam activities, 2) in *keep the status periods*, nodes do not perform any spam check.

A sensor node does not relay unauthenticated messages during the *check the status* period t_c. If it receives an unauthenticated message during t_c, it first requests authentication from the last hop node of the message. If the last hop node fails on authentication, it indicates that the last hop node may be an antinode, therefore the node changes its status to be quarantined. The notion of a quarantine

```
function main()
generate_random( t_c )
reset ( timer_c )
period = check_the_status
status = not_quarantined
in_buffer_zone = false

while ( node is active )
    check_local_alarm_and_buffer_zone_status()
    if ( status == not_quarantined )
        if ( period == keep_the_status )
            if ( timer_k > t_k )
                period = check_the_status
                generate_random( t_c )
                reset ( timer_c )
        else // ( period == check_the_status )
            if ( timer_c > t_c )
                period = keep_the_status
                generate_random( t_k )
                if ( in_buffer_zone == true )
                    t_k = t_k * s // s is keep the status factor
                reset ( timer_k )
            if ( unauthenticated message m is received from node n )
                request authentication of m from n
                if ( authentication of m is unsuccesful )
                    status = quarantined
                    generate_random( t_k )
                    reset ( timer_k )
                    period == keep_the_status
                    if ( no local alarm is sent in the last k * t_k period )
                        broadcast_local_alarm( d )
    else // ( status == quarantined ) and ( period == keep_the_status )
        if ( m is received from n )
            if ( m is not authenticated )
                request authentication of m from n
                if ( authentication of m is unsuccesful )
                    reset ( timer_k )
            if ( timer_k > t_k )
                status = not_quarantined
                generate_random( t_c )
                reset ( timer_c )
                period = check_the_status
end // main

function check_local_alarm_and_buffer_zone_status()
    if ( local alarm is received with depth d )
        generate_random( t_c )
        reset ( timer_c )
        status = not quarantined
        period = check the status
        if ( d > 1 ) and ( no local alarm is sent in the last k * t_k period )
            broadcast_local_alarm( d - 1 )
        in_buffer_zone = true
    if ( in_buffer_zone ) and (time since last local alarm sent > k * t_k )
        in_buffer_zone = false
end // check_local_alarm_and_buffer_zone_status
```

Fig. 1. Quarantine region algorithm.

region is exemplified in Fig. 2. A quarantine region is basically an abstract region where the transmissions of the antinode may be received. This region has an amorphous shape due to the unpredictable propagation environment. The nodes in the quarantine region are the ones that capture the antinode's spam messages during their t_c.

An example sensor field is shown in Fig. 3, where a quarantine region is indicated by the gray area. Nodes 3, 4, 7, 8, and an antinode are in the quarantine region, therefore they have to send and can only relay authenticated messages. Nodes outside the quarantine regions do not need authentication to transmit a message even if the

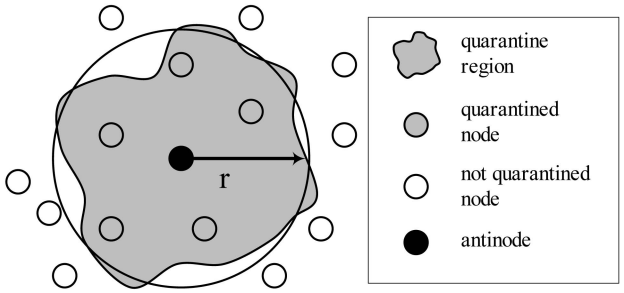


Fig. 2. Boundaries of a quarantine region.

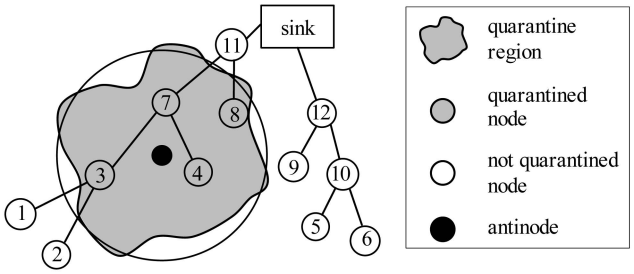


Fig. 3. A sample sensor network and a quarantine region.

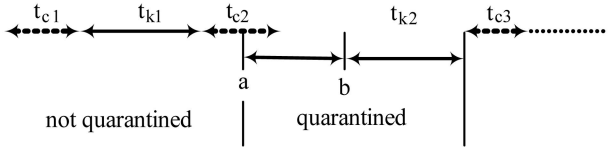


Fig. 4. Timings for the quarantine periods.

message was an originally authenticated message coming from a quarantine region, unless their messages go through a quarantine region. For example, if node 11 receives and authenticated messages from node 7 or 8, it transmits the message unauthenticated towards the sink since that node is not in a quarantine region. On the other hand, if a node in a quarantine region (node 3 in the example) receives an unauthenticated message from a node outside of the quarantine region (node 1 or node 2), it first requests authentication from the corresponding node and relays the data packets only after successful authentication.

QRS is a dynamic scheme where sensor nodes periodically check the validity of their status. If a node does not detect a failure in authentication during the *check the status* period t_c, its status will become *not quarantined*. The node stays in *keep the status* period for t_k and starts another *check the status* period after t_k. If the node detects any authentication failure in the *check the status* period t_c, it changes its status to *quarantined*. This is depicted at point a in Fig. 4, where the node detects an authentication failure and starts a quarantined period. A node starts another *keep the status* period from the very beginning when it detects another authentication failure while in the *keep the status* period. The node exits the quarantined mode and enters into the *check the status* period only if it does not detect any authentication failure during a continuous *keep the status* period.

Both t_c and t_k (i.e., $t_c \geq t_{cmin}, t_k \geq t_{kmin}, t_c \in R^+, t_k \in R^+$) are random variables selected for each period, i.e., t_c and t_k may be different for every *check the status* and *keep the status periods*. Sensor nodes determine these values according to

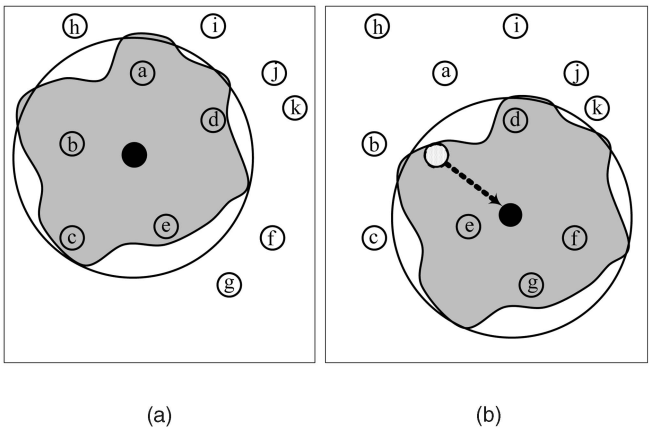


Fig. 5. The change in the boundaries of a quarantine region. (a) Before displacement. (b) After displacement.

the system defined mean values, distribution functions, and t_{cmin}, t_{kmin}. Since antinodes may be mobile or the propagation environment may change temporarily, this dynamic approach is needed.

This procedure and how it changes the location of a quarantine region dynamically is explained by using the illustrative example in Fig. 5. In this example, nodes a, b, c, d, and e independently and asynchronously find out that there is a node within their transmission range that cannot be successfully authenticated by the end of their t_c as shown in Fig. 5a. Therefore, they become quarantined. Nodes f and g are not in the transmission range of the antinode; therefore, they do not detect any authentication failure during their t_c and become *not quarantined*. As shown in Fig. 5b, the antinode moves to a new location which changes its coverage area such that it includes the locations of nodes f and g. Since nodes f and g start a *check the status* period after every *keep the status* period t_k, they find out that they are in the coverage of an antinode in the first *check the status* period. On the other hand, when nodes a, b, and c do not detect any authentication failure for a whole *keep the status* period t_k, they change their status to *not quarantined*.

Spam messages may get through a node during its *keep the status* period when it is not quarantined. *Local alarms* are used to minimize the effects of the spam messages that can access the network due to this reason. A node that detects an authentication failure disseminates a *local alarm* to its d hop neighbors by calling the broadcast_local_alarm function where d is called the *local alarm depth*. As soon as a node receives a *local alarm*, it starts a *check the status* period. By using a local alarm mechanism, the nodes become more alerted and the period that an unsolicited message can get through from a node becomes limited. Local alarm processing is shown in Fig. 1.

Once a node sends a local alarm, it should not send another local alarm for $k \times t_k$ period even if it receives some other spam messages, where k is the local alarm factor, $k \in R^+$. This is done in order not to flood the network with local alarms. An example is depicted in Fig. 6, which shows the timings of local alarms by a node under attack. The node sends a local alarm at point a where it detects an unsolicited message for the first time. Then, it waits at least $k \times t_k$ before sending another local alarm. During this

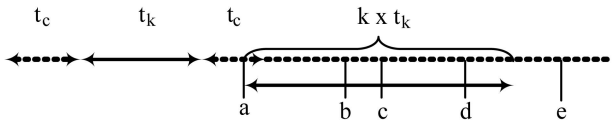


Fig. 6. Timings of the local alarms.

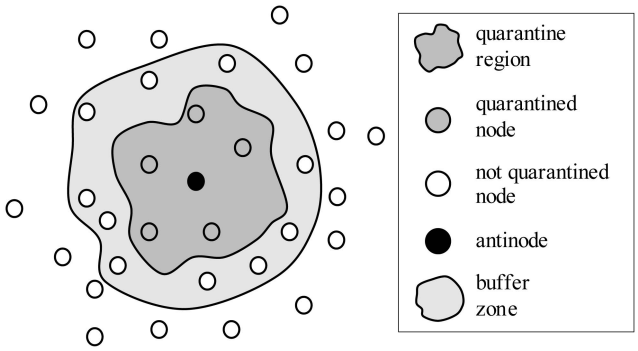


Fig. 7. Buffer zone.

period, it neither generates local alarms for the detected spam messages nor relays the local alarm messages of the other nodes. For example, at points b, c, and d, it detects other unsolicited messages but does not send any local alarms for them. After $k \times t_k$ period, it disseminates another local alarm after the detection of the unsolicited message at Point e.

By local alarms, a buffer zone around the quarantine region is created, as shown in Fig. 7. The nodes in a buffer zone are still *not quarantined*, but are more alert than the nodes that are in a not quarantined region. The node in the buffer zone runs the same algorithm as a *not quarantined* node. It only shortens the *keep the status* period by multiplying it with the *keep the status factor* s, where $0 < s < 1$. A node starts a *check the status* period as soon as it receives a local alarm, which implies that the node is in the buffer zone of a quarantine region. If it does not receive another local alarm during $k \times t_k$, the node assumes that it is not in the buffer zone anymore. The algorithm for buffer zone processing is shown in Fig. 1.

The QRS scheme can also tackle with the antinodes that have long transmission range. This kind of antinode can attempt to attack the entire sensor network from a distance. Our scheme is independent of the location and the transmission range of the antinode. Since the antinode cannot authenticate, its messages are not relayed by the sensor nodes no matter where it is broadcasting its spam messages. In QRS, only the nodes that can receive the transmissions of the antinode carry out the authentication, while the others do not. If there is a node hidden to the antinode due to any reason, it accepts unauthenticated messages even if the antinode transmits with a high power from a long distance because the node does not receive the messages of the antinode. Therefore, the overhead of the authentication is incurred only for the nodes in the coverage area of the antinodes.

2.2 Authentication in a Quarantine Region and Message Format

Authentication in QRS must be simple enough to fit the stringent constraints of tiny sensors. Therefore, we choose

source id	source location	last hop node id	last hop node location	sequence number*
sensed data			authentication code*	

Fig. 8. Authenticated message structure (*: not needed in unauthenticated messages).

message authentication based on cryptographic hash functions. We use the standard HMAC (hash-based message authentication code) mechanism [4], [5] for this purpose. HMAC uses a cryptographic one-way hash function, such as MD5 [6]. The sender and the receiver share a secret key, K. The message authentication code of the message M is calculated as,

$$\mathrm{HMAC} = \mathrm{H}(\mathrm{K} \oplus \mathrm{opad} || \mathrm{H}(\mathrm{K} \oplus \mathrm{ipad} || \mathrm{M})),$$

where $\oplus$ is the bitwise "exclusive OR" operation, H is the underlying one-way hash function, and *ipad* and *opad* are two constants defined in [4], [5].

Power consumption of the HMAC algorithm is an important issue for its usage in sensor networks. We analyzed power consumption for Berkeley Motes [7]. Berkeley motes consume 1 μJ for transmitting and 0.5 μJ for receiving a single bit, while the CPU executes 208 cycles (roughly 100 instructions) with 0.8 μJ [8]. We have implemented the HMAC algorithm in C and assembled it using AVR Studio [9]. We have observed that the HMAC algorithm consumes approximately 45.6 μJ when it is run on a Berkeley mote. This indicates that the overhead of the authentication code generation is equal to approximately 20 percent of the transmission cost for an average QRS packet.

The message fields required for QRS are shown in Fig. 8. *Source id* is the local identification of the source node that generates the *sensed data*. *Last hop node id* is the identification of the last node that relays the message. Every node that relays a message replaces the latter field with its identification. The *last hop node id* is the same as the *source id* when the message is initially transmitted by the source node. *Sensed data* is the payload of the message.

The *source location* and *Last hope node location* fields include the location information according to a coordinate system for the corresponding nodes. Location awareness is generally a requirement and a key point for sensor networks in order to make the data meaningful. Sensor data without location information is accepted to become almost useless, as discussed in [1], [10]. For example, a target detection data is almost meaningless without a location is associated with it. In [1], it has been argued that an ideal sensor network should have attribute-based addressing and location awareness. The possibility of being equipped with a location finding system, together with the requirement of having a highly accurate knowledge of the location of most of the sensor network routing techniques and sensing tasks are also addressed in the same paper. There are various GPS-based, beacon-based, and beaconless location estimation schemes [11], [12], [13] applicable to the tactical sensor networks. Therefore, it is reasonable to assume that the sensor nodes know their location. While GPS may not be used in all sensor networks because of the financial cost [14], other solutions are analyzed in many papers. One solution is offered in [15], where current location sensing technologies, accuracy and precision, scale, cost, and limitations of the techniques are also listed. As localization information is widely used in sensor networks, especially deployed untethered in hostile environments, many attacks have been worked on to prevent processing trustworthy location information. Lazos and Poovendran [16] proposed robust statistical methods to make localization attack-tolerant. Fang et al. [17] proposed a security-aware, range-independent localization scheme.

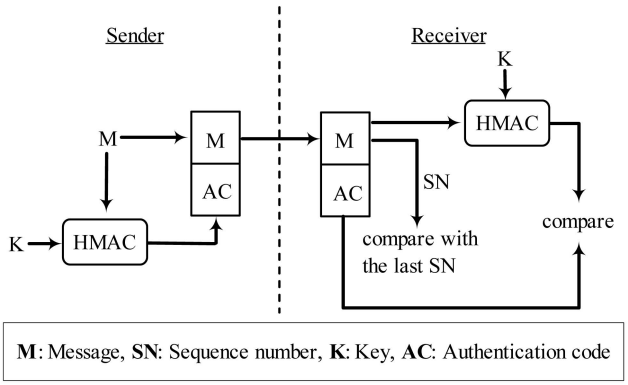


Fig. 9. Message authentication in QRS.

The *sequence number* and *authentication code* fields are added to the message structure in order to support authentication and prevent certain attacks that will be discussed later. The *sequence number* field identifies outgoing messages. The *authentication code* is the calculated HMAC value. These two fields are needed only in authenticated messages; they are not part of the message structure for normal unauthenticated messages.

Please note that these are the fields needed for QRS, which is independent of the transport, network, and medium access layer protocols.

In QRS, we do not focus on key management and key distribution issues. If the network is in a physically inaccessible region and the node compromise is not an issue, then the sensors are equipped with the same secret key K before deployment. If node compromise is considered as a threat, having a single key would cause a problem. In such a case, one of the pairwise key distribution mechanisms [18], [19], [20], [21], [22] described in the literature can be used. These mechanisms also provide resistance against node compromise.

When a sender has a message to send, it first generates the *authentication code* using the HMAC algorithm and the key, K. The message over which HMAC is to be calculated contains the *source id*, *last hop node id*, *sequence number*, and *sensed data* fields. The *sequence number* is incremented for every outgoing message. After the composition, the authenticated message is transmitted. Any node that should relay this message generates the *authentication code* by using the same algorithm, message, and key. If the value calculated at the end of this process is not equal to the value with the *authentication code* field of the incoming message, the message is discarded. Otherwise, the message is accepted. This mechanism is depicted in Fig. 9.

To facilitate the implementation of HMAC in the sensor nodes, (K $\oplus$ opad) and (K $\oplus$ ipad) can be precomputed, as offered in [4] and [5]. This implementation is more efficient, especially when the message is short. As discussed in [23], sensor nodes use small messages (approximately 30 bytes). Thus, this implementation is suitable for QRS.

2.3 More Arguments

QRS thwarts possible replay attacks of antinodes using the sequence numbers. Every sensor node has a counter to be used for *sequence numbers* in outgoing messages. It begins with zero and is incremented by one for each message. Thus, the sequence number of a message created or relayed by a sensor node should be greater than that of every message sent or relayed by the same node before. Each sensor node keeps the last *sequence number* obtained from each of its neighboring nodes. The freshness of each received message is checked by comparing the *sequence number* of the received message with the last *sequence number* of the *last hop node id* of that message that is kept locally. If the received message has a higher *sequence number* and the *authentication code* is verified, then it is concluded that the message is not a replay and is authentic. Such a message is accepted for relaying and the locally kept last *sequence number* is updated accordingly. Relaying nodes do not accept a message with a *sequence number*, which is equal or less than the preceding ones. In such a case, the relaying node asks for the *authentication code* of the same message but with the expected *sequence number*. If the last node cannot regenerate this *authentication code*, then the message is not authenticated.

We assume that the sequence number is long enough that it never repeats within the lifetime of a node. This is an acceptable assumption which also exists in SPINS [23]. Please note that a node needs authentication only for the messages sent when it is in quarantined status and every node generates its own sequence number, i.e., a sequence number can be reused by the other nodes.

One may argue that the *authentication codes* created by a sensor node, which is hidden to another node, $node_a$, can be exploited by antinodes. Since those authenticated messages are not received by $node_a$, the antinodes can record and later resend them to it. $node_a$ accepts those replays as valid and relays them. However, antinodes can never reach to a significant spam rate by using any of these techniques because they need to keep pace with the other nodes to use the *authentication codes* generated by them.

Attackers may collect authenticated and genuine messages from several parts of the network and transfer them to some antinodes in other parts of the network via wormholes [24]. It may be argued that, in this way, antinodes can enable authentication but still perform spam by sending authenticated messages generated in other parts of networks. Sequence numbers help to mitigate such attacks to some extent, but do not solve this problem in its entirety because antinodes may use messages originally generated by nodes that have not sent messages in that region before. In such a case, sensor nodes think that a new node has just moved within its coverage area. Fortunately, *last hop node location* information stored in messages helps to thwart such replay attacks. Whenever a message is to be authenticated, besides MAC verification and sequence number checks, the recipient node also checks if the location of the last hop node is in its coverage area. If the last hop is close enough to the recipient, then the recipient concludes that the message has not been sent out from a distant location. However, the messages used in this attack should originally come from other parts of the network, so they should bear distant location information because, otherwise, the replay attack would become a local one and the recipient would easily spot the attack by checking the sequence numbers as described above. As a side note, readers should notice that antinodes are not capable of changing the *last hop node location* field within the authenticated messages used in the attack because such an attempt would cause nonverifiability of the MACs in the authenticated messages.

3 Performance Metrics

QRS reduces the number of nodes that need authentication when transferring or relaying a data packet, i.e., quarantined nodes. It also limits the time period for authentication such that quarantined nodes can become not quarantined when they are out of the range of any antinode. Therefore, the overhead of the security measures decreases. We call this reduction in the cost of data security "*quarantine gain*" γ. However, a malicious message generated by an antinode may access the sensor network through a sensor node before the sensor node detects that it is in the quarantine region. As the *quarantine gain* γ decreases, the probability that a malicious message is detected by a not quarantined node, i.e., the "*resilience* against an attack" δ, increases. Both *quarantine gain* γ and *resilience* δ are dependent on QRS parameters such as "*check the status*" period t_c, "*keep the status*" period t_k, *antinode range*, and the *malicious message generation rate*. We analyze this relationship in this section analytically.

3.1 Average Quarantine Gain

Nodes do not need to authenticate during *keep the status period* t_k if they are not quarantined. Therefore, the average *quarantine gain* γ is the ratio of t_k to the sum of t_k and *check the status period* t_c. Both t_c and t_k are random variables and the distributions and average values for these variables are selected by the system designers. We prefer uniform distribution for generating these variables because it is easier and computationally more efficient to generate random numbers according to uniform distribution comparing to other continuous distributions. If we assume that *check the status* and *keep the status* periods are uniformly distributed between t_{cmin}, t_{kmin} and t_{cmax}, t_{kmax}, then the average *quarantine gain* γ is

$$\gamma = \psi \int_{t_{c\min}}^{t_{c\max}} \int_{t_{k\min}}^{t_{k\max}} \frac{t_k}{t_k + t_c} dt_k dt_c, \tag{1}$$

where

$$\psi = \frac{1}{(t_{k\max} - t_{k\min})(t_{c\max} - t_{c\min})}.$$

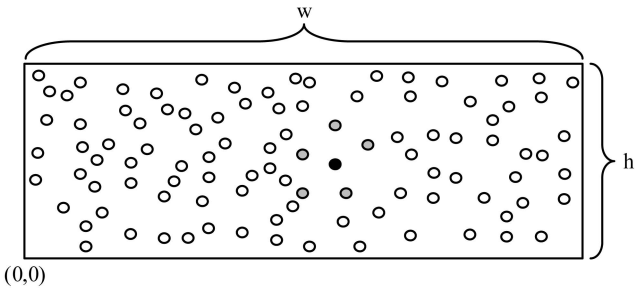


Fig. 10. Sensor network coordinate system.

3.2 Average Resilience against an Attack

The *resilience* δ of a not quarantined node can be given by

$$\delta = 1 - \mathrm{p_n}, \tag{2}$$

where p_n is the probability that a malicious message is not detected by a not quarantined node and can be found out by

$$\mathrm{p_n} = \mathrm{p}_\phi \times \lambda \times \gamma, \tag{3}$$

where p_ϕ is the probability that the not quarantined node is in the range r of the antinode, λ is the malicious message generation rate of the antinode, and γ is the quarantine gain.

The locations in a sensor field can be defined by using a grid coordinate system, as shown in Fig. 10. In this coordinate system, a sensor field is the smallest rectangle that covers all the sensor nodes. The location of the bottommost and leftmost corner of this rectangle is given as 0 in the *X axis* and 0 in the *Y axis*, i.e., from south to north. The width w of the sensor field is the distance between its east and west edges and the height h is the distance between the south and north edges. By using this coordinate system, the probability that a sensor node is in the range of an antinode can be given as

$$p_\phi \;=\; P\left(\sqrt{(x_n - x_e)^2 \,+\, (y_n - y_e)^2} \;\leq\; r\right), \tag{4}$$

where x_n and y_n are the x and y coordinates of the sensor node, x_e and y_e are the x and y coordinates of the antinode, and r is the transmission range of the antinode.

Although the transmission range may not be the same at each direction due to the unpredictable propagation environment, we assume that the antinode has a perfect circular coverage. This is the worst-case scenario for the QRS scheme. In real life, some nodes may be hidden from an antinode even if they are in the transmission range of the antinode. Hence, they will not be quarantined. This will be an additional gain for QRS.

We calculate p_ϕ in two steps: First, we compute the probability density functions (pdf) of $X = x_n - x_e$ and $Y = y_n - y_e$; then, we compute the pdf of $Z = \sqrt{(X^2 + Y^2)}$.

At the first step, by substituting $x_e = x_n - X$ and $y_e = y_n - Y$, we find

$$\begin{aligned} f_X(x) &= \int_{-\infty}^{\infty} f_{X_n X_e}(x_n, x_n - x)dx_e, \\ f_Y(y) &= \int_{-\infty}^{\infty} f_{Y_n Y_e}(y_n, y_n - y)dy_e. \end{aligned} \tag{5}$$

Since x_n, x_e, y_n, and y_e are independent random variables,

$$\begin{aligned} f_X(x) &= \int_{-\infty}^{\infty} f_{X_n}(x_n) f_{X_e}(x_n - x)dx_e, \\ f_Y(y) &= \int_{-\infty}^{\infty} f_{Y_n}(y_n) f_{Y_e}(y_n - y)dy_e. \end{aligned} \tag{6}$$

At the second step, to formulate the pdf of Z, an auxiliary random variable W, as $W = X$, is introduced. This will enable us to use the general formula of finding f_{ZW} by using two functions of two random variables with n real roots, given below:

$$f_{ZW}(z, w) = \sum_{i=1}^{n} f_{XY}(x_i, y_i)\left|\widetilde{J}_i\right|. \tag{7}$$

The equations

$$\begin{aligned} &Z - \sqrt{X^2 + Y^2} = 0, \\ &W - X = 0 \end{aligned} \tag{8}$$

have two real roots, for $|w| < z$, namely,

$$\begin{aligned} x_1 = w \quad & x_2 = w \\ y_1 = \sqrt{z^2 - w^2} \quad & y_2 = -\sqrt{z^2 - w^2} \end{aligned}. \tag{9}$$

At both roots, $\left|\widetilde{J}\right|$ has the same value:

$$\left|\widetilde{J_1}\right| = \left|\widetilde{J_2}\right| = \frac{z}{\sqrt{z^2 - w^2}}. \tag{10}$$

Since X and Y are independent random variables, a direct application of (4) yields

$$f_{ZW}(z, w) = \frac{z}{\sqrt{z^2 - w^2}}[f_X(x_n) f_Y(y_n) + f_X(x_e) f_Y(y_e)]. \tag{11}$$

We get $f_X(x)$ and $f_Y(y)$ in (6), so we can find $F_Z(r)$

$$f_Z(z) = \int_{-\infty}^{\infty} f_{ZW}(z, w)dw, \tag{12}$$

$$P_\phi = F_Z(r) = \int_0^r f_Z(z)dz. \tag{13}$$

Equation (13) gives the probability that a sensor node in the range of an antinode can be extended for the Gaussian distribution, where x_n, x_e, y_n, and y_e are distributed according to $N(0, \sigma_{x_n}^2)$, $N(0, \sigma_{x_e}^2)$, $N(0, \sigma_{y_n}^2)$, and $N(0, \sigma_{y_e}^2)$, respectively, and the Gaussian distribution is centered at the center of the sensor field.

$$\begin{aligned} f_z(z) &= \int_{-\infty}^{\infty} \frac{1}{\pi\sigma^2} \frac{z}{\sqrt{z^2 - w^2}} e^{-z^2/2\sigma^2} dw \\ &= \frac{z}{\sigma^2} e^{-z^2/2\sigma^2} \left[\frac{2}{\pi} \int_0^z \frac{dw}{\sqrt{z^2 - w^2}}\right] u(z). \end{aligned} \tag{14}$$

Since the term in parentheses has value $\pi/2$, $Z = \sqrt{(X^2+Y^2)}, f_z(z)$ is the Rayleigh density function where the standard deviation σ is

$$\sigma = \begin{bmatrix} \sigma_{x_n} - \sigma_{x_e} & 0 \\ 0 & \sigma_{y_n} - \sigma_{y_e} \end{bmatrix}, \tag{15}$$

$$P_\phi = F_Z(r) = \int_0^r \frac{z}{\sigma^2} e^{-z^2/2\sigma^2} d(z). \tag{16}$$

Equation (13) can also be extended for uniform random variables $\mathrm{X_n}(0,w)$, $\mathrm{X_e}(0,w)$, $\mathrm{Y_n}(0,h)$, and $\mathrm{Y_e}(0,h)$, where w and h are the width and height of the sensor field, respectively. If we solve (11) for these random variables, we get

$$f_X(x) = \left\{ \begin{array}{ll} \int_0^{x+w} \frac{1}{w^2} d_{X_n} = \frac{w+x}{w^2}, & -w \le x \le 0 \\ \int_x^w \frac{1}{w^2} d_{X_n} = \frac{w-x}{w^2}, & 0 \le x \le w \end{array} \right\}, \tag{17}$$

$$f_Y(y) = \left\{ \begin{array}{ll} \int_0^{y+h} \frac{1}{h^2} d_{x_n} = \frac{h+y}{h^2}, & -h \le y \le 0 \\ \int_y^h \frac{1}{h^2} d_{y_n} = \frac{h-y}{h^2}, & 0 \le y \le h \end{array} \right\}. \tag{18}$$

The same steps are followed from (13) to (15) and then $f_X(x_1)$, $f_X(x_2)$, $f_Y(y_1)$, and $f_Y(y_2)$ are substituted in (16).

$$f_{ZT}(z,t) = \frac{z}{\sqrt{z^2-t^2}} \left\{ \begin{array}{l} (\frac{w+t}{w^2})(\frac{h+\sqrt{z^2-t^2}}{h^2}) + (\frac{w+t}{w^2})(\frac{h-\sqrt{z^2-t^2}}{h^2}), -w \le x \le 0, -h \le y \le 0 \\ (\frac{w+t}{w^2})(\frac{h-\sqrt{z^2-t^2}}{h^2}) + (\frac{w+t}{w^2})(\frac{h+\sqrt{z^2-t^2}}{h^2}), -w \le x \le 0, 0 \le y \le h \\ (\frac{w-t}{w^2})(\frac{h-\sqrt{z^2-t^2}}{h^2}) + (\frac{w-t}{w^2})(\frac{h+\sqrt{z^2-t^2}}{h^2}), 0 \le x \le w, 0 \le y \le h \\ (\frac{w-t}{w^2})(\frac{h+\sqrt{z^2-t^2}}{h^2}) + (\frac{w-t}{w^2})(\frac{h-\sqrt{z^2-t^2}}{h^2}), 0 \le x \le w, -h \le y \le 0 \end{array} \right\}, \tag{19}$$

$$f_{ZT}(z,t) = \frac{z}{\sqrt{z^2-t^2}} \left\{ \begin{array}{ll} (\frac{2}{h^2})(\frac{1}{w} + \frac{t}{w^2}) & , -w \le x \le 0, -h \le y \le h \\ (\frac{2}{h^2})(\frac{1}{w} - \frac{t}{w^2}) & , 0 \le x \le w, -h \le y \le h \end{array} \right\}, \tag{20}$$

where $z > 0, |t| < z$ conditions must be satisfied.

$$f_Z(z) = \int_{-\infty}^{\infty} f_{ZT}(z,t) dt, \tag{21}$$

$$f(z) = \left\{ \begin{array}{l} \frac{2z}{hw} \int_0^z \frac{dt}{\sqrt{z^2-t^2}} u(z) + \frac{2z}{hw^2} \int_0^z \frac{tdt}{\sqrt{z^2-t^2}} u(z), -w \le x \le 0, -h \le y \le h \\ \frac{2z}{hw} \int_0^z \frac{dt}{\sqrt{z^2-t^2}} u(z) - \frac{2z}{hw^2} \int_0^z \frac{tdt}{\sqrt{z^2-t^2}} u(z), -w \le x \le 0, -h \le y \le h \end{array} \right\} \tag{22}$$

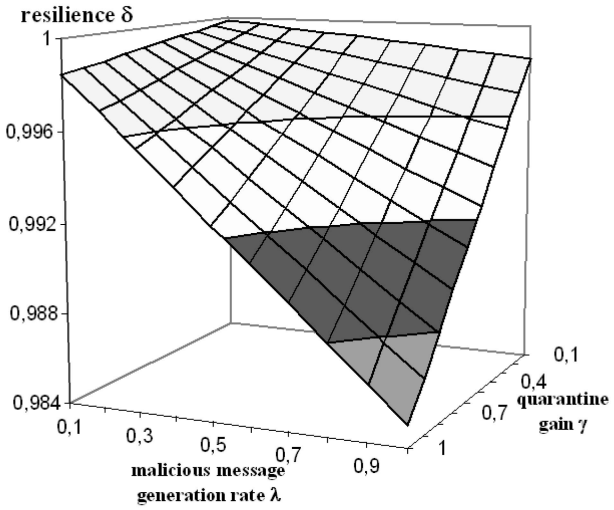


Fig. 11. Resilience versus malicious message generation rate and quarantine gain.

if we substitute v with $v = z^2 - t^2$, so dv becomes $dv = -2wdw$ and, by solving the integrals, since $z \ge 0$ and $|w| < z$, we get:

$$f_Z(z) = \left\{ \begin{array}{l} \left(\frac{2z}{hw}\left[\arcsin\frac{t}{z}\right]_0^z + \frac{2z}{hw^2}\left[-\sqrt{z^2-t^2}\right]_0^z\right) u(z), -w \le x \le 0, -h \le y \le h \\ \left(\frac{2z}{hw}\left[\arcsin\frac{t}{z}\right]_0^z - \frac{2z}{hw^2}\left[-\sqrt{z^2-t^2}\right]_0^z\right) u(z), -w \le x \le 0, -h \le y \le h \end{array} \right\}, \tag{23}$$

$$f_Z(z) = \left\{ \begin{array}{ll} \frac{2z}{hw}(\frac{\pi}{2} + \frac{z}{w}) u(z), & -w \le x \le 0, -h \le y \le h \\ \frac{2z}{hw}(\frac{\pi}{2} - \frac{z}{w}) u(z), & -w \le x \le 0, -h \le y \le h \end{array} \right\}. \tag{24}$$

The probability that a sensor node is in the *range* r of an antinode becomes:

$$P_\phi = F_Z(r) = \int_0^r f_Z(z) dz, \tag{25}$$

$$P_\phi = \left\{ \begin{array}{ll} \frac{\pi r^2}{2hw} + \frac{2r^3}{3hw^2}, & -w \le x \le 0, -h \le y \le h \\ \frac{\pi r^2}{2hw} - \frac{2r^3}{3hw^2}, & 0 \le x \le w, -h \le y \le h \end{array} \right\}. \tag{26}$$

By using (16) or (26), we can extend (3) and then (2), which gives the average resilience δ against an attack for a not quarantined node.

4 The Experimental Results

In this section, the performance of QRS in the percentage of authenticated data packets, i.e., *quarantine gain* γ, and the percentage of spam packets that cannot access the network, i.e., *resilience* δ, is evaluated both by the mathematical models in Section 3 and through simulation.

The resilience δ of the QRS found out by (2) is depicted in Figs. 11 and 12 for various antinode transmission *range* r, malicious message generation *rate* λ, and *quarantine gain* γ. In Fig. 11, the value of antinode transmission *range* r is 10

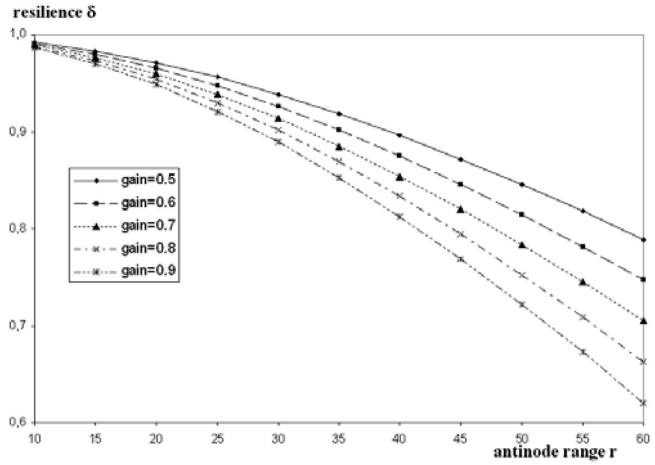


Fig. 12. Resilience versus antinode range and quarantine gain.

and the sensor field size is 100 both in *height* h and *width* w. In this case, more than 3 percent of the sensor nodes receive the transmissions created by the antinode when sensor nodes are deployed according to Uniform distribution. In Fig. 12, the sensitivity of resilience against the antinode transmission range is examined. When the antinode range r is 60, the antinode covers an area larger than the sensor field, which implies that the transmissions of the antinode are received by almost every node in the sensor field. Even in this case, more than 80 percent of spam messages are detected by sensor nodes when the quarantine gain is 50 percent and the probability that there is a spam message in the air at any moment is 0.5, as shown in Fig. 12. When the coverage area of the antinode decreases, the probability that a spam message is detected, i.e., resilience δ, can be as high as over 0.99, as shown in Fig. 12.

In our simulations, we randomly deploy 100 sensor nodes and 10 antinodes in a sensor field 200×200 in size according to Uniform distribution. One of the topologies used in our simulation is shown in Fig. 13. In the figure, the sensor nodes are depicted with "+" and the antinodes with "*". Sensor nodes are fixed. However, antinodes move in the sensor field according to the random waypoint model. The average speed of the antinodes is factored in simulations.

The data generation rate of sensor nodes is 0.01 packets per second (pps) and the malicious packet generation rate

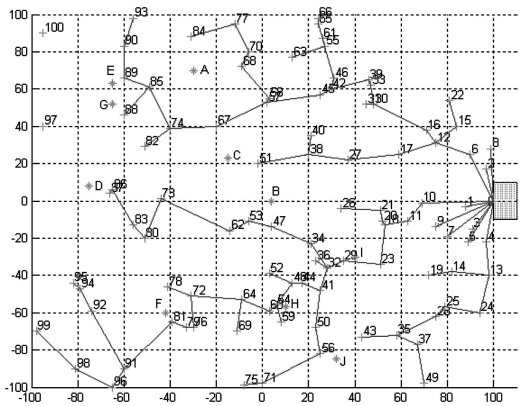


Fig. 13. Sensor and antinode deployment in a sensor field.

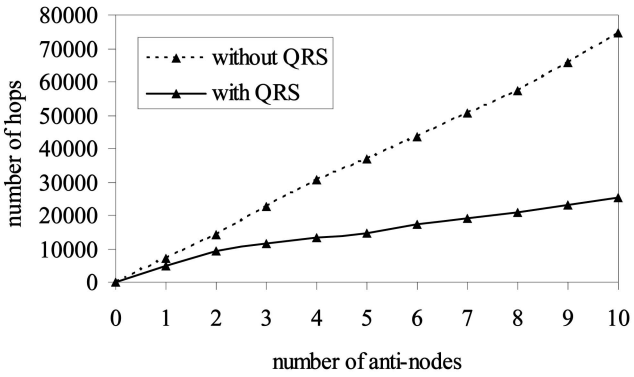


Fig. 14. Effect of QRS in a sensor field with 100 sensor nodes and some antinodes.

of antinodes is 0.5 pps. Since the change in the packet generation rate of sensor nodes does not have any impact on the results, it is fixed and not randomized. This also helps us in being fair and analyzing the results easier.

In Fig. 14, we depict how the QRS scheme prevents spam traffic effectively by reducing the total number of hops or the network traffic caused by antinodes in the network. We simulate the sensor network with one to 10 antinodes, each generating 100 messages. When analyzing the situation, we have run our simulations using 10 different random network setups similar to the one shown in Fig. 13 and depicted the average behavior in Fig. 14. The QRS scheme eliminates 66 percent of the traffic caused by antinodes.

In Fig. 15, we depict the effect of spam frequency in QRS with 10 antinodes in the network. Since it takes some time to detect the existence of a spam attack, network performances are very close to each other when the number of spam messages per antinode is small. The percentage of traffic eliminated by the QRS scheme increases as the number of messages generated by antinodes increases.

We also examine the sensitivity of QRS for varying antinode transmission *range* r, local alarm *depth* n, local alarm *factor* k, and quarantine *gain* γ. Note that the quarantine gain γ is a function of *keep the status period* t_k and *check the status period* t_c. We use directed diffusion [25] as the data dissemination scheme. Since physical layer parameters, such as path loss exponent and hop distance, do not affect our results, they are not factoring parameters in our experiments.

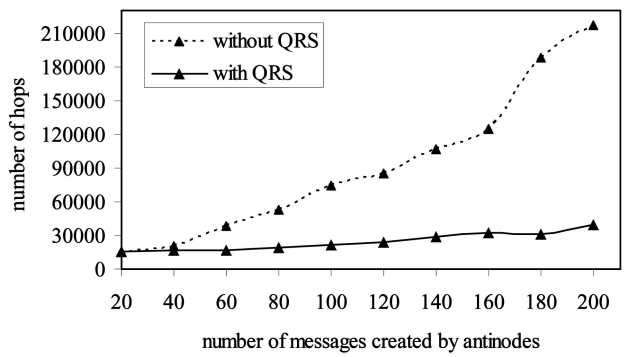


Fig. 15. Impact of spam frequency.

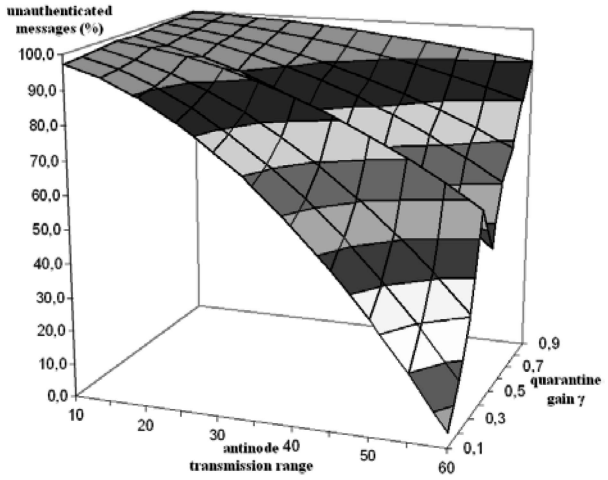


Fig. 16. Percentage of unauthenticated messages versus antinode transmission *range* r and quarantine *gain* γ.

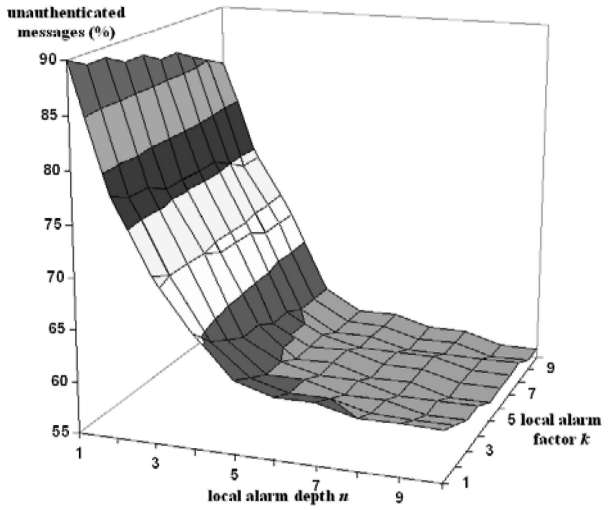


Fig. 17. Percentage of unauthenticated messages versus local alarm *factor* k and *depth* n for quarantine *gain* $\gamma = 0.5$ and node *range* $r = 25$.

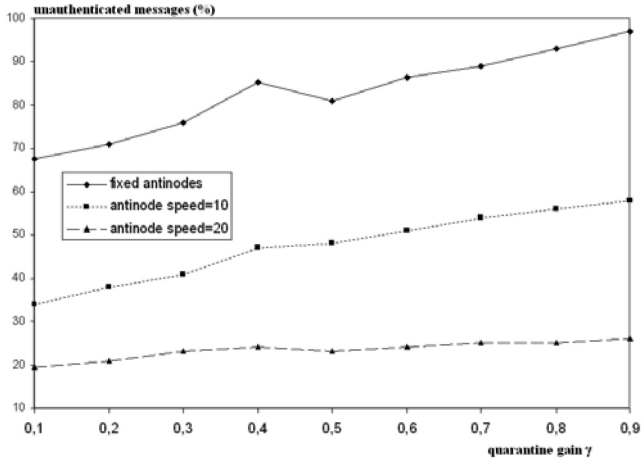


Fig. 18. Percentage of unauthenticated messages versus quarantine *gain* γ and antinode mobility.

In Figs. 16, 17, and 18, we examine the ratio between the number of unauthenticated transmissions and the total number of transmissions (the unauthenticated message percentage), which is more than 80 percent for quarantine gain $\gamma > 0.5$ and the antinode transmission *range* $r < 30$ as shown in Fig. 16. This indicates that only 20 percent of the messages are authenticated even when the sensor field is completely under attack of antinodes. The impact of local alarm *depth* n on the unauthenticated message percentage is higher comparing to the impact of *local alarm factor* k as shown in Fig. 17. The sensitivity of the unauthenticated message percentage against the mobility of antinodes is depicted in Fig. 18. When antinode speed is 20 meters per minute, it can travel from one side of the sensor field to the other side in 10 minutes. In such a case, the percentage of the unauthenticated messages is as low as 20 because almost every node in the sensor network that we simulate is in a quarantine region continuously.

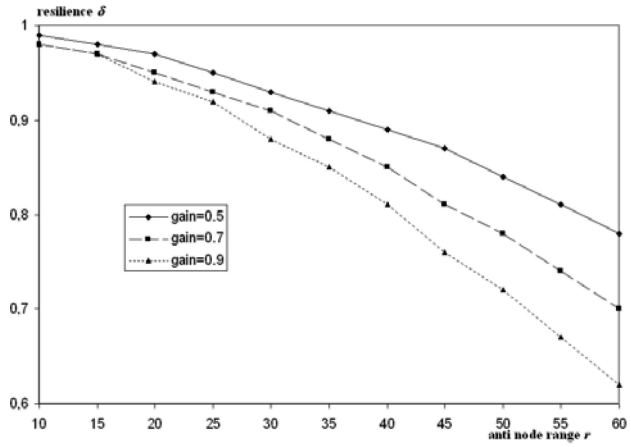


Fig. 19. Percentage of the malicious messages that can access the sensor network versus quarantine *gain* γ and antinode *range* r.

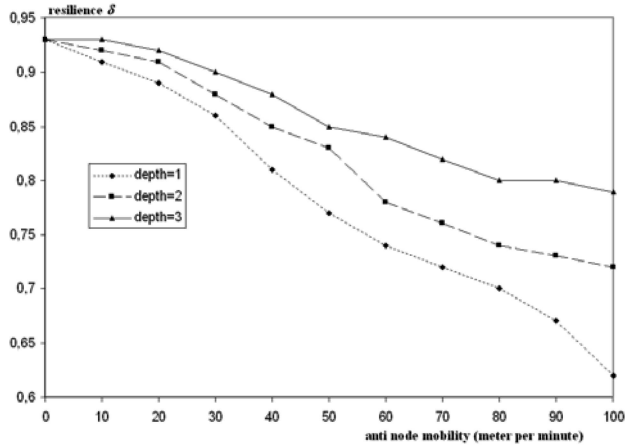


Fig. 20. Percentage of the malicious messages that can access the sensor network versus quarantine *gain* γ and antinode *range* r.

In Fig. 19, the resilience δ is shown for varying antinode range. The values in this figure are from our simulations, and they are almost the same as the values in Fig. 12, which are found out by the analytical models in Section 3. This also verifies our mathematical models.

In Fig. 20, the resilience δ is shown for varying antinode speeds when quarantine *gain* $\gamma = 0.7$ and antinode *range* $r = 25$. When the antinode speed is 100 meters per minute, an antinode can travel from one side of our sensor field to the other side in two minutes. Please note that we have 10 antinodes in our sensor field, which means almost every

TABLE 1
Number of Nodes within Transmission Range, r = 25

Number of neighbors within range	0	1	2	3	4	5	6	7	8	9	10	11	Total
Number of nodes having that much neighbors	3	7	10	17	14	13	11	10	4	6	4	1	100

TABLE 2
Size of the Adjacency List for r = {10..50}

Range	10	15	20	25	30	35	40	45	50
Maximum size of Adjacency List	3	5	7	11	13	15	21	24	29
Average size of Adjacency List	0,72	1,52	2,98	4,72	6,48	8,38	11,46	14,24	17,28

node is under attack continuously. Even in this extreme case, more than 80 percent of the malicious messages are eliminated by our scheme in our simulations.

Since the storage capacity of sensor nodes is limited, we also examine the storage requirement that QRS imposes on each node. We have stated in Section 2.3 that each sensor node has to keep the last *sequence number* received from each *adjacent node*, i.e., nodes within the transmission range; we call this list as the *adjacency list*. Here, the readers should remark that only the last sequence number from each adjacent node needs to be stored; no history is to be stored. This requirement may raise a question as to how much storage load is imposed by the QRS scheme. We have calculated the number of nodes within the transmission range, $r = 25$, of each node by using the simulation setup which was depicted in Fig. 13. The results are shown in Table 1. As shown in this table, the length of list per node varies between 0 and 11, with the average length of the list being 4.72. By looking at these values, we may easily conclude that the QRS scheme does not impose hard requirements on the sensor nodes for keeping sequence numbers of adjacent nodes.

We also calculated the size of adjacency list with different ranges {10..50}. The maximum and average sizes of the adjacency list are shown in Table 2. Here, notice that the highest transmission range in the table, $r = 50$, is too big for a 200×200 sensor network, whereas, even in this case, the maximum adjacency list size is 29 and the average adjacency list size is 17.28, which are still not very high. This also verifies our previous argument that QRS scheme does not impose too much overhead for keeping the sequence number information.

5 Related and Complementary Work

Perrig et al. give a good overview of security-related challenges and attacks in wireless sensor networks in [26]. Various security schemes that tackle these types of attacks are introduced for ad hoc networks in [27], [28], and [29]. However, they are not well suited for the features and application requirements of sensor networks due to the differences between ad hoc and sensor networks [1].

Spam attacks that we studied in this paper may be considered as a kind of Denial of Service (DoS) attack since the aim of antinodes is to halt the sensor network. Layer by layer vulnerabilities and defense mechanisms to DoS attacks for a typical sensor network are discussed by Wood and Stankovic in [3].

As discussed in [3], attempts to add DoS resistance to existing protocols often focus on cryptographic-authentication mechanisms. However, there are also some noncryptographic DoS resistant systems proposed in the literature. Marti et al. [30] proposed two extensions to DSR (Dynamic Source Routing) [31] in order to detect and mitigate routing misbehavior. These extensions are *watchdog* and *pathrater*. The watchdog detects misbehaving nodes, while the pathrater avoids routing packets through these nodes. The watchdog scheme has further been extended, yielding rating schemes [32], [33], [34]. In the rating scheme, the neighbors of a particular node collaborate in rating the node according to how well the node executes the functions requested.

Besides DoS, several other types of attacks on sensor networks are identified and some countermeasures are proposed by Karlof and Wagner in [2] and Hu et al. in [35]. Among these attacks, Sybil [36] and wormhole [24] are especially related to the attack model defined in this paper. In Sybil attacks, an attacker presents multiple identities to other nodes. Antinodes can do the same in our attack model, but they are caught by sensor nodes during authentication. In wormhole attacks, attackers tunnel messages received in one part of network to another part and replay there. The wormhole problem in QRS is mentioned and a countermeasure is proposed in Section 2.3.

Restricting the damage caused by a compromised node to its local area is a similar aim with keeping an antinode's effect limited to the quarantine region. Deng et al. proposed and evaluated the performance of the INSENS (Intrusion Tolerant Routing in Wireless Sensor Networks) scheme [37], [38] that has the property that a single compromised node can only disrupt a localized portion of the network.

False data injection to the sensor network is another type of malicious node attack. In the literature, there are some solutions proposed for this problem using the selective authentication approach, which is a similar approach to QRS. Ye et al. [39] propose a statistical en-route detection scheme, which allows en-route detection of false data packets. A recent study by Zhu et al. [40] proposes an interleaved hop-by-hop authentication scheme that guarantees that the base station detects any injected false packets. Independent of this study, Vogt [41] also proposes an interleaved authentication approach for data integrity protection.

SPINS (Security protocols for sensor networks) is one of the security schemes proposed for sensor networks [23]. In SPINS, two secure building blocks are used: SNEP (Secure Network Encryption Protocol) and μTESLA (the micro version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol). SNEP provides data confidentiality, two party data authentication, and data freshness. μTESLA provides an authenticated streaming broadcast.

Establishing message authentication in sensor networks is the main idea behind fighting against spam attacks. In that respect, SPINS also fights against spam. However, in SPINS, unlike QRS, all packets need to be authenticated.

Vogt [41] explores several types of message authentication techniques, including end-to-end, hop-by-hop, and multipath authentication, and gives some design space considerations for message authentication in sensor networks.

QRS may be seen as a sort of exclusion scheme, where the antinodes are isolated from the rest of the network. Another interesting exclusion study for wireless sensor networks is given by Di Pietro et al. in [42], where they introduce the concept of *Supervisor*, which can invoke the secure exclusion of a malicious node from the network using a scalable and cooperative algorithm that also provides rekeying.

There are several key distribution schemes proposed in the literature [18], [19], [20], [21], [22], [43]. Most of them [18], [19], [20], [21], [43] are based on the random key predistribution idea adopted from [20]. Moreover, most of the proposed key distribution schemes [18], [19], [21], [22], [43] provide resistance against node compromise to some extent. In QRS, we focused on intermittent authentication patterns of sensor nodes and assumed that the necessary key for authentication is predeployed in nodes. However, QRS may be combined with a key distribution and management scheme in order to assign different keys to different nodes and links and in order to make the system resistant against node compromise while avoiding spam attacks.

6 Conclusion

In this paper, we introduce QRS (Quarantine Region Scheme) for defending against spam attacks performed by hostile antinodes scattered around the field. In the scope of QRS, we propose spam detection and location-aware message authentication mechanisms. The spam detection mechanism is a totally distributed one. Each node decides whether there is a spam attack in its reception range by requesting authenticated messages in some irregular and random time periods. Thus, the antinodes cannot synchronize themselves with the legitimate nodes and eventually get caught. A node that detects a spam attack in its vicinity puts itself in a quarantine region and performs continuous message authentication until the attack comes to an end. As the above discussions imply, in QRS, message authentication is not performed all the time. Consequently, the overhead of authentication measures is reduced and limited according to the size of the threat.

The dynamic features of spam detection and location-aware message authentication mechanisms help the system to survive even where the sensor and antinodes are mobile. Moreover, the system becomes resistant against various kinds of replay and wormhole attacks.

Analytical models for performance evaluation are also developed. QRS is evaluated both by using these models and through simulations. The results prove the gains and effectiveness of QRS. Our analysis also shows that QRS is a trade-off scheme such that it is possible to play with some parameters in order to set a level for the trade-off between the resilience against spam attacks and the number of authentications. Such a trade-off could be useful to play according to several system factors, such as sensitivity of the field against spam attacks, the remaining battery power of sensor nodes, etc.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks J.,* vol. 38, no. 4, pp. 393-422, 2002.

[2] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Ad Hoc Networks,* vol. 1, nos. 2-3, pp. 293-315, Sept. 2003.

[3] A.D. Wood and J.A. Stankovic, "Denial of Service in Sensor Networks," *Computer,* vol. 35, pp. 54-62, 2002.

[4] W. Stallings, *Cryptography and Network Security,* third ed. Prentice Hall, 2003.

[5] H. Krawczyk, M. Bellare, and R. Canetti, "RFC 2104—HMAC: Keyed-Hashing for Message Authentication," 1997.

[6] R.L. Rivest, "RFC 1321—The MD5 Message-Digest Algorithm," 1992.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Network Sensors," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems,* 2000.

[8] S. Bhattacharya, H. Kim, S. Prabh, and T.F. Abdelzaher, "Energy-Conserving Data Placement and Asynchronous Multicast in Wireless Sensor Networks," *Proc. Int'l Conf. Mobile Systems, Applications, and Services,* 2003.

[9] http://www.atmel.com/products/AVR, Oct. 2003.

[10] J. Beutel, "Location Management in Wireless Sensor Networks, Handbook of Sensor Networks," *Compact Wireless and Wired Sensing Systems,* July 2004.

[11] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," *Proc. First ACM Workshop Wireless Sensor Networks and Applications,* pp. 105-111, 2002.

[12] A. Savvides, H. Park, and M.B. Srivastava, "The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems," *Proc. First ACM Workshop Wireless Sensor Networks and Applications,* pp. 112-121, 2002.

[13] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks," *Proc. Int'l Symp. Information Processing in Sensor Networks,* 2005.

[14] G. Amato, A. Caruso, S. Chessa, V. Masi, and A. Urpi, "State of the Art and Future Directions in Wireless Sensor Network's Data Management," *Istituto di Scienza e Tecnologie dell'Informazione del CNR,* May 2004.

[15] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer,* vol. 34, no. 8, pp. 57-66, Aug. 2001.

[16] L. Lazos and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," *Proc. ACM Workshop Wireless Security,* 2004.

[17] L. Fang, W. Du, and P. Ning, "A Beacon-Less Location Discovery Scheme for Wireless Sensor Networks," *Proc. IEEE Infocom,* 2005.

[18] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," *Proc. IEEE Symp. Security and Privacy,* pp. 197-213, May 2003.

[19] W. Du, J. Deng, Y. Han, and P. Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," *Proc. 10th ACM Conf. Computer and Comm. Security,* Oct. 2003.

[20] L. Eschenauer and V. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proc. Ninth ACM Conf. Computer and Comm. Security,* Oct. 2002.

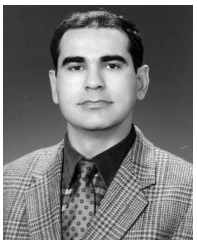
[21] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," *Proc. 10th ACM Conf. Computer and Comm. Security,* Oct. 2003.

[22] R. Di Pietro, L.V. Mancini, and S. Jajodia, "Providing Secrecy in Key Management Protocols for Large Wireless Sensors Networks," *Ad Hoc Networks,* vol. 1, no. 4, pp. 455-468, Nov. 2003.

[23] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J.D. Tygar, "SPINS: Security Protocols for Sensor Networks," *Proc. Int'l Conf. Mobile Computing and Networks (MobiCom '01),* pp. 189-199, 2001.

[24] Y.C. Hu, A. Perrig, and D.B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks," *Proc. IEEE Infocom,* Mar. 2003.

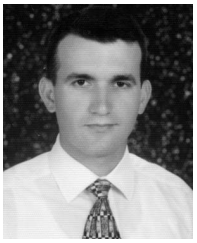
[25] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. Sixth Ann. Int'l Conf. Mobile Computing and Networks (MobiCom '00)*, 2000.
[26] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *Comm. ACM*, vol. 47, no. 6, pp. 53-57, June 2004.
[27] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks," *Proc. Seventh Int'l Workshop Security Protocols*, pp. 172-182, 2000.
[28] L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24-30, 1999.
[29] J.P. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," *Proc. ACM Symp. Mobile Ad Hoc Networking and Computing*, 2001.
[30] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. Sixth Ann. Int'l Conf. Mobile Computing and Networking (MobiCom '00)*, pp. 255-265, 2000.
[31] D.B. Johnson, D.A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," *Ad Hoc Networking*, C.E. Perkins, ed., chapter 5, pp. 139-172, Addison-Wesley, 2001.
[32] P. Michiardi and R. Molva, "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *Proc. Comm. and Multimedia Security Conf.*, 2002.
[33] P. Michiardi and R. Molva, "Prevention of Denial of Service Attacks and Selfishness in Mobile Ad Hoc Networks," Research Report RR-02-063, Institut Eurecom, 2002.
[34] P. Michiardi and R. Molva, "Simulation-Based Analysis of Security Exposures in Mobile Ad Hoc Networks," *Proc. European Wireless 2002: Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Feb. 2002.
[35] F. Hu, J. Ziobro, J. Tillett, and N.K. Sharma, "Secure Wireless Sensor Networks: Problems and Solutions," *J. Systemics, Cybernetics, and Informatics*, vol. 1, no. 4, 2004.
[36] J.R. Douceur, "The Sybil Attack," *Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '02)*, 2002.
[37] J. Deng, R. Han, and S. Mishra, "INSENS: Intrusion-Tolerant Routing in Wireless Sensor Networks," Technical Report CU-CS-939-02, Dept. of Computer Science, Univ. of Colorado, Nov. 2002.
[38] J. Deng, R. Han, and S. Mishra, "A Performance Evaluation of Intrusion Tolerant Routing in Wireless Sensor Networks," *Proc. Second IEEE Int'l Workshop Information Processing in Sensor Networks (IPSN 2003)*, pp. 349-364, Apr. 2003.
[39] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Detection and Filtering of Injected False Data in Sensor Networks," *Proc. IEEE Infocom*, 2004.
[40] 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," *Proc. IEEE Symp. Security and Privacy*, pp. 260-272, 2004.
[41] H. Vogt, "Exploring Message Authentication in Sensor Networks," *Proc. First European Workshop Security in Ad Hoc and Sensor Networks*, Aug. 2004.
[42] R. Di Pietro, L.V. Mancini, and S. Jajodia, "Secure Selective Exclusion in Ad Hoc Wireless Network," *Security in the Information Society: Visions and Perspectives*, M.A. Ghonaimy, M.T. El-Hadidi, and H.K. Aslan, eds., pp. 423-434, Kluwer Academic, 2002.
[43] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pairwise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach," *Proc. 11th IEEE Int'l Conf. Network Protocols (ICNP '03)*, Nov. 2003.



Vedat Coskun (M'02) graduated from the Turkish Naval Academy (1984) and received the MSc degree in computer science from the Naval Post Graduate School, California (1990), and the PhD degree in computer security from Yildiz Technical University, Istanbul (1998). He managed the software development group at the Naval Wargaming Center between 1997-2001. He was chairman of the Computer Engineering Department, Turkish Naval Academy, and is currently with ISIK University, Istanbul, Turkey. His current research areas include software engineering, sensor networks, computer security, and cryptography. He is a member of the IEEE.

Erdal Cayirci (M'96-SM'04) graduated from the Turkish Army Academy in 1986 and from the Royal Military Academy Sandhurst in 1989. He received the MS degree from the Middle East Technical University and the PhD degree from Bogaziçi University in computer engineering in 1995 and 2000, respectively. He retired from the Turkish Army when he was a Colonel in 2005. He is currently the chief of the CAX Support Branch at NATO's Joint Warfare Center. He is also with the Electrical and Computer Engineering Department at the University of Stavanger. His research interests include military constructive simulation, tactical communications, sensor networks, and mobile communications. He was an editor for the *IEEE Transactions on Mobile Computing*, *AdHoc Networks* (Elsevier Science), *ACM/Kluwer Wireless Networks*, and *ASP Sensor Letters*. He received the 2002 IEEE Communications Society Best Tutorial Paper Award for his paper titled "A Survey on Sensor Networks" published in *IEEE Communications Magazine* in August 2002 and the "Fikri Gayret" Award from the Turkish Chief of General Staff in 2003. He is a senior member of the IEEE and a member of the IEEE Computer Society.

Albert Levi (S'96-M'00) received the BS, MS, and PhD degrees in computer engineering from Bogaziçi University, Istanbul, Turkey, in 1991, 1993, and 1999, respectively. He served as a visiting faculty member in the Department of Electrical and Computer Engineering, Oregon State University between 1999 and 2002. He was also a postdoctoral research associate in the Information Security Lab of the same department. He worked at rTrust Inc. as a consultant in 2001-2002. Since 2002, he has been a faculty member of computer science and engineering at Sabanci University, Faculty of Engineering and Natural Sciences, Istanbul, Turkey, and codirector of the Cryptography and Information Security (CISEC) Lab. His research interests include computer and network security with an emphasis on mobile and wireless system security, public key infrastructures (PKI), and application layer security protocols. He is a member of the IEEE and the IEEE Computer Society.



Serdar Sancak received the BS degree in computer engineering from the Turkish Naval Academy in 1998 and received the MS degree in computer science from the Naval Science and Engineering Institute in 2003. Currently, he is the director of information systems at Northern Sea Area Command. His research interests include sensor networks, network security, and wireless communications.