# Security Mechanisms for Wireless Sensor Networks

*Xiuli Ren,*[1, 2] *and Haibin Yu*[1]

Shenyang Institute of Automation, Chinese Academy of Science, 110016 Shenyang, China
School of Computer Jilin Normal University, 136000 Siping, China

**Summary**

Wireless sensor networks have been identified as being useful in a variety of domains to include military sensing and tracking, environment monitoring, patient monitoring and tracking smart environment, etc. When sensor networks are deployed in a hostile environment, security becomes extremely important, as they are prone to different types of malicious attacks. Due to the resource limitations of sensor nodes, existing network security mechanisms, including those developed for Mobile Ad-Hoc Networks, are inadequate for wireless sensor networks. In this paper, we give some security mechanisms to adapt to wireless sensor networks for sensor data and network control protocols.

***Key words:***
*Wireless sensor networks, Security, Mechanism, Key.*

## 1. Introduction

Recent advances in electronic and computer technologies have paved the way for the proliferation of wireless sensor networks. Sensor networks usually consist of a large number of ultra-small autonomous devices. Each device, called a sensor node, is battery powered and equipped with integrated sensors, data processing capabilities, and short-range radio communications. In typical application scenarios, sensor nodes are spread randomly over the deployment region under scrutiny and collect sensor data . Wireless sensor networks are being deployed for a wide variety of applications [1], including military sensing and tracking, environment monitoring, patient monitoring and tracking, smart environments, etc. When sensor networks are deployed in a hostile environment, security becomes extremely important, as they are prone to different types of malicious attacks. For example, an adversary can easily listen to the traffic, impersonate one of the network nodes, or intentionally provide misleading information to other nodes. To provide safe data, communication should adopt security mechanisms.

Wireless sensor network distinguishes itself from other traditional wireless networks by relying on extremely constrained resources like energy, bandwidth and capabilities of processing and storing data. Traditional security techniques used in traditional networks can not be applied directly, and new ideas are need. In this paper, we give some security mechanisms to adapt to wireless sensor networks.

The rest of the paper is organized as follows. In Section II we categorize possible threats and analyze the security needs in the wireless sensor networks. Section III gives some security mechanisms. Section IV concludes the paper and points out the future research direction.

## 2. Security Threats and Analysis

### 2.1 Threats

Wireless networks, in general, are more vulnerable to security attacks than wired networks, due to the broadcast nature of the transmission medium. Furthermore, wireless sensor networks have an additional vulnerability because nodes are often placed in a hostile or dangerous environment where they are not physically protected. For data sent through the network, the main security threats are as follows:

- Insertion of malicious code is the most dangerous attack that can occur. Malicious code injected in the network could spread to all nodes, potentially destroying the whole network, or even worse, taking over the network on behalf of an adversary. A seized sensor network can either send false observations about the environment to a legitimate user or send observations about the monitored area to a malicious user.
- Interception of the messages containing the physical locations of sensor nodes allows an attacker to locate the nodes and destroy them. The significance of hiding the location information from an attacker lies in the fact that the sensor nodes have small dimensions and their location cannot be trivially traced. Thus, it is important to hide the locations of the nodes. In the case of static nodes, the location information does not age and must be protected through the lifetime of the network.
- Besides the locations of sensor nodes, an adversary can observe the application specific content of messages including message IDs, timestamps and other fields. Confidentiality of those fields in the application is less important than confidentiality of location information, because the application specific data does not contain

Manuscript received March 25, 2006.
Manuscript revised March 30 , 2006.

sensitive information, and the lifetime of such data is significantly shorter.

- An adversary can inject false messages that give incorrect information about the environment to the user. Such messages also consume the scarce energy resources of the nodes. This type of attack is called sleep deprivation torture in [2].

### 2.2 Analysis

In this section we discuss the major security concerns in wireless sensor networks and their corresponding requirements.

Confidentiality: Unauthorized parties should not be able to infer the content of messages. Due to the shared wireless medium, the adversary can eavesdrop on the messages exchanged between sensor nodes. To prevent the release of message content to eavesdroppers, efficient cryptographies can be used for message encryption before transmissions.

Integrity: The receiver should be able to detect any modifications to a received message during its transmission. This prevents, for example, man-in-the-middle attacks where an adversary overhears, alters, and re-broadcasts messages. By including message authentication codes (MAC), a cryptographically strong un-forgeable hash, with the packet, the packet integrity can be protected. Using a secret key for code generation, unauthenticated nodes will not be able to alter the content of legitimate messages in the network.

Authentication: Message authentication is important for many applications in sensor networks. Within the building sensor network, authentication is necessary for many administrative tasks (e.g. network reprogramming or controlling sensor node duty cycle). At the same time, an adversary can easily inject messages, so the receiver needs to make sure that the data used in any decision-making process originates from the correct source. Informally, data authentication allows a receiver to verify that the data really was sent by the claimed sender. In the two-party communication case, data authentication can be achieved through a purely symmetric mechanism: The sender and the receiver share a secret key to compute a message authentication code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that it must have been sent by the sender.

Access Control: Unauthorized nodes should not be able to participate in the network by either acting as a router or injecting new traffic. By including message authentication code (MAC) with the packet, unauthenticated nodes will not be able to send legitimate messages into the network.

Semantic security: Semantic security ensures that an eavesdropping adversary can not obtain information about the plaintext, even if it sees multiple encryptions of the same message. The lack of semantic security makes traffic analysis easy. One common method of achieving this in symmetric block cipher is to use an Initial Value in the encryption function; this value may be a random value sent with the message or kept implicitly by both parties as a counter or the clock value.

Message replay protection: Even if messages are cryptographically protected so that their contents cannot be inferred or forged, an attacker would be able to capture valid messages and replay them later. Thus, independence on what mechanism is selected to secure the messages, that mechanism must be protected against replay attacks. Replay protection guarantees the system is immune to the stale or falsely located information. Generally, replay attacks can be defeated at the price of network synchronization and additional communication overhead.

Freshness: Given that all sensor networks stream some forms of time varying measurements, it is not enough to guarantee confidentiality and authentication; we also must ensure each message is fresh. Informally, data freshness implies that the data is recent, and it ensures that no adversary replayed old messages. Two types of freshness are identified: weak freshness, which provides partial message ordering, but carries no delay information, and strong freshness, which provides a total order on a request-response pair, and allows for delay estimation. Weak freshness is required by sensor measurements, while strong freshness is useful for time synchronization within the network.

## 3. Security Mechanisms

The security of wireless sensor networks has attracted a lot of attention in the recent years. Many researchers have proposed some security mechanisms. In the section, we primarily introduce several ones.

### 3.1 Localized Encryption and Authentication Protocol (LEAP)

LEAP provides multiple keying mechanisms that can be used for providing confidentiality and authentication in sensor networks. It supports the establishment of four types of keys for each sensor node – an individual key shared with the base station, a pairwise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a group key that is shared by all the nodes in the network. Now each of these keys is discussed and established in the LEAP protocol.

### 3.1.1 Type of Key

Individual Key: Every node has a unique key that it shares pairwise with the base station. This key is used for secure communication between a node and the base station. For example, a node may send an alert to the base station if it observes any abnormal or unexpected behavior by a neighboring node. Similarly, the base station can use this key to encrypt any sensitive information, e.g. keying material or special instruction, it sends to an individual node.

Cluster Key: A cluster key is a key shared by a node and all its neighbors, and it is mainly used for securing locally broadcast messages, e.g. routing control information, or securing sensor messages which can benefit from passive participation. Researchers have shown that in-network processing techniques, including data aggregation and passive participation are very important for saving energy consumption in sensor networks [3, 4, 5]. For example, a node which overhears a neighboring sensor node transmitting the same reading as its own current reading can elect not to transmit the same. In responding to aggregation operations such as MAX, a node can also suppress its own reading if its reading is not larger than an overheard one. For passive participation to be feasible, neighboring nodes should be able to decrypt and authenticate some classes of messages, e.g. sensor readings, transmitted by their neighbors. This means that such messages should be encrypted or authenticated by a locally shared key. Therefore, in LEAP each node possesses a unique cluster key that it uses for securing its messages, while its immediate neighbors use the same key for decryption or authentication of its messages.

Pairwise Shared Key: Every node shares a pairwise key with each of its immediate neighbors. In LEAP, pairwise keys are used for securing communications that require privacy or source authentication. For example, a node can use its pairwise keys to secure the distribution of its cluster key to its neighbors, or to secure the transmissions of its sensor readings to an aggregation node. Note that the use of pairwise keys precludes passive participation.

Group Key: This is a globally shared key that is used by the base station for encrypting messages that are broadcast to the whole group. For example, the base station issues missions, sends queries and interests. Note that from the confidentiality point of view there is no advantage to separately encrypting a broadcast message using the individual key of each node. However, since the group key is shared among all the nodes in the network, an efficient re-keying mechanism is necessary for updating this key after a compromised node is revoked.

### 3.1.2 Key Establishment

Individual Keys: Every node has an individual key that is only shared with the base station. This key is generated and pre-loaded into each node prior to its deployment. The individual key $k_u^m$ for a node *u* (each node has a unique ID) is generated as follows: $k_u^m = f_{k_s^m}(u)$. Here *f* is a pseudo-random function and $k_s^m$ is a master key known only to the controller. In this scheme the controller might only keep its master key to save the storage for keeping all the individual keys. When it needs to communicate with an individual node *u*, it computes $k_u^m$ on the fly. Due to the computational efficiency of pseudo random functions, the computational overhead is negligible.

Cluster Keys: The cluster key establishment phase follows the pairwise key establishment phase, and the process is very straightforward. Consider the case that node *u* wants to establish a cluster key with all its immediate neighbors $v_1, v_2, ..., v_m$. Node *u* first generates a random key $k_u^c$, then encrypts this key with the pairwise key of each neighbor, and then transmits the encrypted key to each neighbor $v_i$.

$$u \rightarrow v_i : (k_u^c)_{k_{uv_i}} .$$

Node $v_i$ decrypts the key $k_u^c$ and stores it in a table. When one of the neighbors is revoked, node *u* generates a new cluster key and transmits to all the remaining neighbors in the same way.

Pairwise Shared Key: A pairwise shared key belonging to a node refers to a key shared only between the node and one of its direct neighbors (i.e. one-hop neighbors). For nodes whose neighborhood relationships are predetermined (e.g. via physical installation), pairwise key establishment is simply done by preloading the sensor nodes with the corresponding keys. The protocol establishes pairwise keys for sensor nodes unaware of their neighbors until their deployment (e.g. via aerial scattering). The approach exploits the special property of sensor networks consisting of stationary nodes that the set of neighbors of a node is relatively static, and that a sensor node that is being added to the network will discover most of its neighbors at the time of its initial deployment. Second, it is that a sensor node deployed in a security critical environment must be designed to sustain possible break-in attacks at least for a short interval (say several seconds) when captured by the adversary; otherwise, the adversary could easily compromise all the sensor nodes in a sensor network and then take over the network.

Group Key: A group key is a key shared by all the nodes in the network, and it is necessary when the controller is distributing a secure message, e.g. a query on some event of interest or a confidential instruction, to all the nodes in the network. One way for the base station to distribute a message *M* securely to all the nodes is using hop-by-hop translation. Specifically, the base station encrypts *M* with its cluster key and then broadcasts the message. Each neighbor receiving the message decrypts it to obtain *M*, re-encrypts *M* with its own cluster key, and then re-broadcasts the message. The process is repeated until all the nodes receive *M*. However, this approach has a major drawback, that is, each intermediate node needs to encrypt and decrypt the message, thus consuming a non-trivial amount of energy on computation. Therefore, using a group key for encrypting a broadcast message is preferable from the performance point of view. A simple way to bootstrap a group key for a sensor network is to pre-load every node with the group key. An important issue that arises immediately is the need to securely update this key when a compromised node is detected. In other words, the group key must be changed and distributed to all the remaining nodes in a secure, reliable and timely fashion. The naive approach in which the base station encrypts the updated group key using the individual key of each node and then sends the encrypted key to each node separately is not scalable because its communication and computational costs increase linearly with the size of the network. The protocol proposes an efficient key updating scheme based on cluster keys: authentic node revocation and secure key distribution.

## 3.2 Random Key Predistribution Schemes

The main phases for random key predistribution schemes [6, 7, 8, 9] are as follows:

Key predistribution phase: A centralized key server generates a large key pool offline. The procedure for offline key distribution is as follows:

- Assign a unique node identifier or key ring identifier to each sensor.
- Select *m* different keys for each sensor from the key pool to form a key ring.
- Load the key ring into the memory of the sensor.

Sensor deployment phase: The sensors are randomly picked and uniformly distributed in a large area. Typically, the number of neighbors of a sensor (*n*) is much smaller than the total number of deployed sensors (*N*).

Key discovery phase: During the key discovery phase, each sensor broadcasts its key identifiers in clear-text or uses private share-key discovery scheme to discover the keys shared with its neighbors. By comparing the possessed keys, a sensor can build the list of reachable nodes with which share keys and then broadcast its list. Using the lists received from neighbors, a sensor can build a key graph (see Definition 1) based on the key-share relations among neighbors.

Pairwise key establishment phase: If a sensor shares key(s) with a given neighbor, the shared key(s) can be used as their pairwise key(s). If a sensor does not share key(s) with a given neighbor, the sensor uses the key graph built during key discovery phase to find a key path (see Definition 2) to set up the pairwise key. The set of all neighbors of sensor *i* is represented by $W_i$. The definition of key graph is given as follows:

Definition 1 (key graph). A key graph maintained by node *i* is defined as $G_i = (V_i , E_i)$ where, the vertices set $Vi = \{j | j \in W_i \vee j = i\}$, the edges set $E_i = \{e_{jk} | j, k \in W_i \wedge j R k \}$, *R* is a relation defined between any pair of nodes *j* and *k* if they share required number of key(s) after the key discovery phase.

Definition 2 (key path). A key path between node *A* and *B* is defined as a sequence of nodes $A, N_1, N_2,. . ., N_i, B$, such that, each pair of nodes $(A, N_1)$, $(N_1, N_2)$, . . ., $(N_{i-1}, N_i)$, $(N_i ,B)$ has required number of shared key(s) after the key discovery phase. The length of the key path is the number of pairs of nodes in it.

### 3.2.1 Purely Randoom Key Predistribution (P-RKP)

There are two characteristics of current P-RKP schemes. First, the *m* keys preinstalled in a sensor can also be installed in other sensors. That is, a key can be shared by more than one pair of sensors. Second, in most of current schemes, there is no relation between the set of preloaded keys and the sensor ID. A recent solution proposed by Pietro et al. [10] attempts to define this relation. However, the scheme is not scalable in that the size of the network is restricted by a function of number of preinstalled keys.

### 3.2.2 Structured Key Pool Random Key Predistribution (SK-RKP) Scheme

Unlike in P-RKP schemes, in SK-RKP scheme, each sensor is preloaded with a unique set of keys in its memory. The key discovery is not simply finding a shared key with the neighboring sensor, but using a set of polynomial variables (constructed by the keys possessed by the sensor) to derive the shared key. In addition, the key ID can serve as the sensor ID which is linked to the set of preinstalled keys. This link can prevent the attackers from misusing the sensors' IDs. In the following paragraphs, a brief description of structured key pool scheme is given. The SK-RKP scheme uses the key predistribution scheme proposed by Blom [11]. This scheme allows any pair of nodes in a network to find a pairwise key in a secure way as long as no more than λ nodes are compromised. The scheme is built on two matrices: a publicly known matrix *G* of size $(\lambda + 1) \times N$; a secret matrix *D* of size $(\lambda + 1) \times (\lambda + 1)$ created by key distribution center. The matrix *A* of size $N \times (\lambda + 1)$ is then created as $A = (D \cdot G)^T$ . Each row of $A$ is the keys distributed to a group member and the row number can serve as a sensor's ID. Since $K = A \cdot G$ is a symmetric matrix, nodes $i$ and $j$ can generate a shared key ($K_{ij}$ or $K_{ji}$ ) from their predistributed secrets, where $K_{ij}$ is the element in $K$ located in the $i$th row and $j$th column.

A key pool is constructed by many key spaces, represented by $A^{(t)}$, where $t = 1, \ldots, \omega$. Each sensor randomly selects $\tau$ key spaces out of $\omega$ key spaces, where $\tau < \omega$. If sensor $k$ selects key space $A^{(t)}$, the $k$th row of $A^{(t)}$ and $k$th column of $G$ are preinstalled in the sensor (note that the $G$ matrix is unique). The SK-RKP scheme has following properties:

- Once two nodes $i$ and $j$ have keys presinstalled from the same key space $A^{(t)}$, they can derive a shared key $K_{ij}^{(t)} = K_{ji}^{(t)}$ .
- If $x$ rows of a key space $A^{(t)}$ are predistributed to $x$ sensors and $x \leq \lambda$ , any subset of the $x$ sensors cannot collude to derive the secrets in other sensors.
- The ID of a sensor is represented by the row number of the key matrix $A$. No other sensor can impersonate this sensor, since the row of $A$ is uniquely distributed to this sensor.

## 3.3 Security Levels Based on Different Data

The mechanism for communication security in wireless sensor networks is that data items must be protected to a degree consistent with their value. There are three types of data sent through the network: mobile code, locations of sensor nodes and application specific data. Following this categorization, the three security levels described here are based on private key cryptography utilizing group keys. Since all three types of data contain more or less confidential information, the content of all messages in the network is encrypted. The mechanism is assumed that all sensor nodes in the network are allowed to access the content of any message.

The deployment of security mechanisms in a sensor network creates additional overhead. Not only does latency increases due to the execution of the security related procedures, but also the consumed energy directly decreases the lifetime of the network. To minimize the security related costs, following the taxonomy of the types of data in the network, three security levels are defined:

- Security level I is reserved for mobile code, the most sensitive information sent through the network.
- Security level II is dedicated to the location information conveyed in messages.
- Security level III is applied to the application specific information.

The strength of the encryption for each of security levels corresponds to the sensitivity of the encrypted information. Therefore, the encryption applied at level I is stronger than the encryption applied at level II, while the encryption on level II is stronger than the one applied at level III. Different security levels are implemented either by using various algorithms or by using the same algorithm with adjustable parameters that change its strength and corresponding computational overhead. Using one algorithm with adjustable parameters has the advantage of occupying less memory space. RC6 [12] is selected. It is suitable for modification of its security strength because it has an adjustable parameter (number of rounds) that directly affects its strength. The overhead for the RC6 encryption algorithm increases with the strength of the encryption measured by the number of rounds.

### 3.3.1 Security Level I

The messages that contain mobile code are less frequent than the messages that the application instances on different nodes exchange. It allows us to use a strong encryption in spite of the resulting overhead. For information protected at this security level, nodes use the current master key. The set of master keys, the corresponding pseudorandom number generator, and a seed are credentials that a potential user must have in order to access the network. Once when the user obtains those credentials, she can insert any code into the network. If a malicious user breaks the encryption on this level using a "brute force" attack, she can insert harmful code into the network.

### 3.3.2 Security Level II

For data that contains locations of sensor nodes, a novel security mechanism is provided which isolates parts of the network, so that breach of security in one part of the network does not affect the rest of the network.

According to the applications expected to run in sensor networks, the locations of sensor nodes are likely to be included in the majority of messages. Thus, the overhead that corresponds to the encryption of the location information significantly influences the overall security overhead in the network. This must be taken into account when the strength of the encryption at this level is determined. Since the protection level is lower for the location information than for mobile code, the probability that the key for the level II can be broken is higher. Having the key, an adversary could potentially locate all nodes in the network. To constrain the damage to only one part of the network, the following security mechanism is proposed. Sensor nodes use location-based keys for level II encryption. The location-based keys enable separation between the regions where the location of nodes are compromised and the areas where nodes continue to operate safely.

The area covered by a sensor network is divided into cells. Nodes within one cell share a common location-based key, which is a function of a fixed location in the cell and the current master key. Between the cells, there is a bordering region whose width is equal to the transmission range. Nodes belonging to those regions have the keys for all adjacent cells. This ensures that two nodes within a transmission range from each other have a common key. The dimensions of the cells must be big enough so that the localized nature of the algorithms in the network ensures that the traffic among the cells is relatively low, compared to overall traffic. The areas can be of an arbitrary shape with the only requirement that the whole sensor terrain is covered. A division of the area in uniformly sized cells is the most appropriate solution, because it allows a fast and easy way for a node to determine its cell membership. The network is divided into hexagonal cells, since it ensures that the gateway nodes have at most three keys.

3.3.3 Security Level III

The application specific data use a weaker encryption than the one used for the two aforementioned types of data. The weaker encryption requires lower computational overhead for application specific data. Additionally, the high frequency of messages with application specific data prevents using stronger and resource consuming encryption. Therefore, an encryption algorithm that demands less computational resources with a corresponding decrease in the strength of security is adopted.

The key used for the encryption of the level III information is derived from the current master key. The MD5 hash function accepts the master key and generates a key for level III. Since the master key is periodically changed, the corresponding key at this level follows those changes.

In the discussion above the major assumptions of the all the proposed security schemes is that the sensor nodes are perfectly time synchronized and have exact knowledge of their location. It is not unrealistic that the nodes can be synchronized up to µs.

## 4. Conclusion and Future Work

Security in wireless sensor networks has attracted a lot of attention in the recent years. In this paper, some security mechanisms are introduced. To some extent, they can satisfy the need of security for the wireless sensor networks. But the severe constraints and demanding deployment environments of wireless sensor networks make computer security for these systems more challenging than for conventional networks. To achieve a secure system, security must be integrated into every component, since components designed without security can become a point of attack. Consequently, security and privacy pervade every aspect of system design. Ongoing direction is how to secure wireless communication links against eavesdropping, tampering, traffic analysis, and denial of service.

### Acknowledgments

This work was supported by National Natural Science Project under grant 60374072 and by National High Tech Research Development Project under grant 2004AA41-2020.

## References

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] F. Stajano, R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", 3rd AT&T Software Symposium, Middletown, NJ, October 1999.

[3] C.In tanagonwiwat, R.Go vindan and D.Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks In Proc. of MobiCOM'00, Boston, Massachussetts, August 2000.

[4] C.Karlof, Y.Li, and J.P olastre.ARRIVE: An Architecture for Robust Routing In Volatile Environments.T echnical Report UCB/CSD-03-1233, University of California at Berkeley, Mar.2003.

[5] S.Madden, R.Szew czyk, M.F ranklin, and D.Culler. Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks.In 4th IEEE Workshop on Mobile Computing Systems & Applications, June 2002.

[6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of 9th ACM Conference on Computer and Communication Security (CCS-02), November 2002, pp. 41–47.

[7] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in Proceedings of 2003 Symposium on Security and Privacy. Los Alamitos, CA: IEEE Computer Society, May 11–14 2003, pp. 197–215.

[8] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03), October 2003, pp. 52–61.

[9] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03), October 2003, pp. 42–51.

[10] R. D. Pietro, L. V. Mancini, and A. Mei, "Efficient and resilient key discovery based on pseudo-random key pre-

deployment," in 18th International Parallel and Distributed Processing Symposium (IPDPS'04), April 2004.

[11] R. Blom, "An optimal class of symmetric key generation systems," in EUROCRYPT'84, ser. Lecture Notes in Computer Science, vol. 209. Paris, France: Springer-Verlag, 1985, pp. 335–338.

[12] R. L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, "The RC6 Block Cipher", AES submission, Jun 1998. http://theory.lcs.mit.edu/~rivest/rc6.pdf.

Xiuli Ren received the M.S. and Ph.D. degrees in Computer Application Technology from School of Information Science & Engineering, Northeastern University in 1999 and 2004, respectively. During 1999-2004, she studied protocols and algorithms of wireless networks communication at all times. Now she is engaged in study of wireless networks security as a postdoctoral researcher.