Protection of an Intrusion Detection Engine with Watermarking in Ad Hoc Networks

Aikaterini Mitrokotsa[1], Nikos Komninos[2], and Christos Douligeris[3]
(Corresponding author: Aikaterini Mitrokotsa)

Information and Communication Theory Group, Faculty of Electrical Engineering[1]
Mathematics and Computer Science, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
Algorithms and Security Group, Athens Information Technology, GR-19002 Peania Greece[2]
Department of Informatics, University of Piraeus, GR-18534, Piraeus, Greece[3]
(Email: A. Mitrokotsa@TUDelft.nl, nkom@ait.edu.gr, cdoulig@unipi.gr)
(Received Nov. 19, 2008; revised and accepted Feb. 11, 2009)

Abstract

In this paper we present an intrusion detection engine comprised of two main elements; firstly, a neural network for the actual detection task and secondly watermarking techniques for protecting the related information that must be exchanged between nodes. In particular, we exploit information visualization and machine learning techniques in order to achieve efficient and effective intrusion detection. In order to avoid possible modification or alteration of the maps produced by the intrusion detection engine, we focus on safeguarding and authenticating them using a novel embedded watermarking method. Previously, we had shown promising results in the intrusion detection task using this system. This paper focuses on the watermarking technique and gives a detailed exposition that includes an experimental evaluation of its quality.

Keywords: Intrusion detection, mobile Ad Hoc networks, network security, watermarking

1 Introduction

Wireless communications are being adopted in a broad range of environments making the need for the rapid deployment of various wireless networking technologies essential. One of these technologies, the Mobile Ad hoc NETworks (MANET), also called spontaneous networks, consist of a collection of mobile nodes, which employ a multi-hop information transfer without relying on an *a priori* infrastructure [3, 4, 25]. In a MANET all nodes communicate in a self-organized way and they may appear or disappear from the network at any time. The mobile devices create a wireless communication channel, in which each one of them contributes to the routing decisions of the network, as well as other basic network services. Mobile nodes communicate directly with nodes in their vicinity and they use intermediate nodes in order to exchange information with nodes out of their radio range. Effective cooperation is thus required for good performance.

Although MANET are very flexible, they also present a number of inherent vulnerabilities that pose unique security requirements and consequently research challenges [15, 17, 19]. For instance, MANET's topology is changing dynamically and are susceptible to numerous threats including passive eavesdropping, spoofing and modification of information.

Intrusion prevention mechanisms can be used to reduce possible intrusions but they cannot eliminate them. Wired networks have long used intrusion detection as a second line of defense, but its deployment in MANET is still in its infancy. An efficient combination of intrusion prevention and detection mechanisms is necessary in order to reliably and efficiently safeguard MANET. In this paper we present an intrusion detection module as part of a local IDS architecture composed of a data collection engine, an intrusion detection engine and a response engine. The paper examines the combination of machine learning, information visualization and watermarking techniques, focusing on the latter. We propose an intrusion detection approach based on a type of neural network model called eSOM (emergent Self-Organizing Maps), which is distributed among nodes. We combine this with a novel watermarking technique, which is used to thwart the alteration of the model when it is communicated between the nodes.

More specifically, each node of the MANET creates a map that depicts its security state and distributes this map to all its neighboring nodes. Thus, each node knows the security status of every neighbor by generating a global map. The global map is used to securely and efficiently route data, by avoiding paths that include com-

promised nodes. Watermarking techniques are applied to protect the produced maps from modification. The combined watermarking technique derives from the *Lattice* and *Block-Wise* [9, 26, 33] methods. When local maps are broadcasted to the neighboring nodes, a cryptographic encoder and decoder can authenticate them. Using eSOM, each node can determine whether a neighboring node is under attack and forward its messages accordingly.

Watermarking techniques have advantages [16, 32] that are not provided by simple encryption. Firstly, they can be used by each node to prove ownership of its eSOM map. In addition, they provide copy protection. This means that it is impossible for an intruder to copy an eSOM map and present it as its own. Furthermore, modification in watermarked eSOM maps can be easily detected. Thus, using watermarking techniques we can authenticate the nodes of a MANET, verify the integrity of the maps produced by eSOM and provide copy protection of the eSOM maps. In our previous work [20] we proposed an efficient intrusion detection engine based on eSOM and a reliable intrusion response engine. We improve upon this by safeguarding the output of the Intrusion Detection engine from possible alteration or substitution, through the use of a reliable authentication mechanism: the proposed watermarking technique.

Following this introduction, the paper is organized as follows. Section 2 presents related work of intrusion detection approaches that have been proposed for mobile ad hoc networks and approaches that use watermarking techniques. Section 3 discusses the intrusion detection model this paper is based on. Section 4 presents a functional description of the proposed detection engine and the classification algorithm used. Section 5 presents the watermarking technique proposed for the authentication of maps produced by eSOMs. In Section 6 the performance evaluation of the detection engine as well as the results of the proposed watermarking technique are presented. Finally, Section 7 concludes the paper and discusses future work.

1.1 Related Work

Intrusion Detection [2, 24] is an active and mature research area in wired networks but techniques designed for wired networks may not be efficient if applied to wireless ad hoc networks due to the stringent requirements these networks present. Compared to wired networks where traffic monitoring is performed in gateways, routers and switches, wireless ad hoc networks lack centralized choke points at which it would be possible to monitor network traffic. Even if we could achieve the existence of such concentration points, their locations would continuously change due to mobility. This is the reason why the deployment of a distributed intrusion detection approach in wireless ad hoc networks is a necessity. Additionally, we should focus on security mechanisms keeping in mind the ease of listening to wireless transactions, the lack of a fixed infrastructure and the resource consumption characteristics of MANET. This means that it is better to use a periodic intrusion detection system (IDS) than an "always-on" prevention mechanism. Moreover, the resource constraints that MANET face including limited battery capabilities, strict bandwidth requirements and frequent miscommunication complicate the discrimination between a new qualified operation after a disconnection and an intrusion, a fact that makes even more difficult the classification between normal and attack behavior.

The architecture that will be used for applying the intrusion detection system has been a challenging issue for research. The architecture of an IDS applied to MANET could be either distributed and cooperative or distributed and hierarchical. The distributed and hierarchical IDSs are based on dividing the mobile ad hoc network in clusters. Zhang and Lee [33] proposed the first (high-level) IDS approach specific for ad hoc networks. They proposed a distributed and cooperative anomaly-based IDS, which provides an efficient guide for the design of IDSs in wireless ad hoc networks. They focused on an anomaly detection approach based on routing updates on the MAC layer and on the mobile application layer. Deng et al. [7], Liu et al. [18], Tseng et al. [27], Chen et al. [5] and Anjum et al. [1] adopted the distributed architecture of the intrusion detection system and used various classification methods as well as data from different layers.

More specifically, Deng et al. [7] proposed a hierarchically distributed and a completely distributed intrusion detection approach. The intrusion detection approach used in both of these architectures focuses on the network layer and it is based on a Support Vector Machines (SVM) classification algorithm. They used a set of parameters derived from the network layer and suggested that a hierarchically distributed approach may be a more promising solution versus a completely distributed intrusion detection approach. Liu et al. [18] proposed a completely distributed anomaly detection approach. They investigated the use of the MAC layer in order to profile normal behavior of mobile nodes and then applied cross-feature analysis [12] on feature vectors constructed from the training data.

Tseng et al. [27] proposed a distributed and specification-based intrusion detection approach in order to detect attacks in the AODV (Ad hoc On-Demand Distance Vector Routing) routing protocol. The approach involves the use of finite state machines. More specifically, correct AODV routing behavior is specified using finite state machines and the actual behavior of AODV flows is compared with these specifications. Any deviation from these specifications is recognized as intrusion. Specification-based techniques have the drawback that it is necessary to balance the trade-off between complexity and accuracy. Chen et al. [5] proposed a distributed intrusion detection approach based on the Dempster-Shafer theory. They exploited the main advantages of this theory and its ability to reflect uncertainty or a lack of complete information and the convenient numerical procedure for fusing together multiple pieces of data.

Anjum et al. [1] proposed a signature-based intrusion detection approach for wireless ad hoc networks based on the assumption that attack signatures are completely known in an ad hoc network. This approach investigates the ability of various routing protocols to facilitate the intrusion detection procedure. The authors show that reactive ad-hoc routing protocols are less effective than proactive routing protocols in the detection of intrusions even in the absence of mobility.

On the other hand, Huang and Lee [13] and Kachirski and Guha [14] proposed the use of a cluster based architecture as more efficient for implementing intrusion detection in wireless ad hoc networks. More specifically, Huang and Lee [13] extended their previous work by proposing a cluster-based IDS, in order to combat the resource constraints that MANET face. They used a set of statistical features that can be derived from routing tables and they applied the classification decision tree induction algorithm C 4.5 in order to detect normal versus abnormal behavior. The proposed system is able to identify the source of the attack, if the identified attack occurs within one hop.

Kachirski and Guha [14] proposed a cluster-based intrusion detection system built on a mobile agent framework. The proposed system uses mobile agents each performing a particular role, either monitoring, or decision or action. A few nodes are chosen by a distributed algorithm in order to host sensors for the monitoring of network packets and agents in order to make the decisions. Additionally, all the nodes host sensors for host-based monitoring. The main advantage of this approach is that the packet-monitoring task is limited in a few nodes and the IDS-related processing time by each node is minimized.

Although cluster-based IDSs have the advantage of lower detection workload, the procedure of creating clusters and electing cluster heads may cause a great overhead. Moreover, the existence of cluster heads and the obvious possibility of their exploitation by malicious attackers might act as single point of failure. Furthermore, the distributed hierarchical IDSs are more efficient for ad hoc networks with low mobility. Thus, the cooperative and dynamic nature of MANET implies that the intrusion detection system should be distributed and cooperative. The lack of central monitoring nodes and the lack of trust between peer nodes of a wireless ad hoc network render a central intrusion detection system impractical.

Furthermore, all the previous approaches are based on anomaly or signature based approaches in order to implement intrusion detection and notification through alarm messages. Considering the fact that intrusion detection in wireless ad hoc networks should be performed in real time and have a response as quickly as possible, we exploit the visual representation that eSOM can provide us. We use eSOM in order to classify and discriminate normal and attack behavior. Thus, we are able to visualize and interact with the produced map representation. Maps provide us efficient ways to navigate and expand, since they can give us a unique perception of space. However, in the field of network security we still watch activities in cyberspace through a keyhole [10]. Maps that represent network traffic can help us significantly in order to acquire a global view about the security status of wireless ad hoc networks. With the proposed approach we are able to use the produced map metaphor in order to have a clear view of the secure nodes in a wireless ad hoc network and to select the appropriate ones for forwarding messages.

However, in order to use the important advantages that information visualization provide us, we have to be sure that the visual representations will not be altered or modified by malicious users. An efficient technique to safeguard visual representations is watermarking. Watermarking is a mature research area that has been used extensively in the research area of information security. More specifically, in the area of intrusion detection Wang et al. [31] proposed a framework for intrusion detection in wired networks where watermarking and tracing of the packets to the attacker's source IP address is activated when the IDS subsystem determines that there is an attack in progress.

Páez et al. [23] proposed a security scheme for Intrusion Detection Systems based on Cooperative Itinerant Agents (CIA). They proposed a new security scheme in order to verify the entities' integrity of an Intrusion Detection System based on mobile cooperative agents using watermarking software techniques. More specifically, they proposed the use of fingerprinting software in order to differentiate agents of the same kind and to detect more sophisticated attacks.

Despite the important advantages that watermarking techniques [7, 32] present no application of watermarking techniques in the area of securing ad hoc networks has been proposed. In this paper, we use watermarking in combination with eSOM in order to ensure that the exploitation of the information visualization that eSOM provide will not be altered by malicious attackers. In MANET the response to possible attacks should be quick considering the resource constraints that they face. Information visualization can help us in order to have a direct response in possible intrusions. With the availability of the proposed scheme each node has the option, when choosing where to forward its information, to select a secure neighbor node and not one that can be a likely subject of an attack.

2 Intrusion Detection Model

Malicious nodes in a MANET may target to exploit features of the physical, MAC and/or network layers. The majority of the so far proposed security approaches in such networks has focused in the network layer, while little research has been done on the MAC layer security. The role of the MAC layer in wireless ad hoc networks is substantial as it is responsible for maintaining the communication between nodes and the scheduling of the access in a shared radio channel. The MAC layer is directly affected by almost every intrusion [8, 18], since it is placed

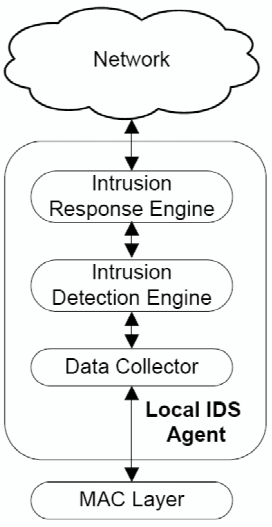

Figure 1: Intrusion detection architecture

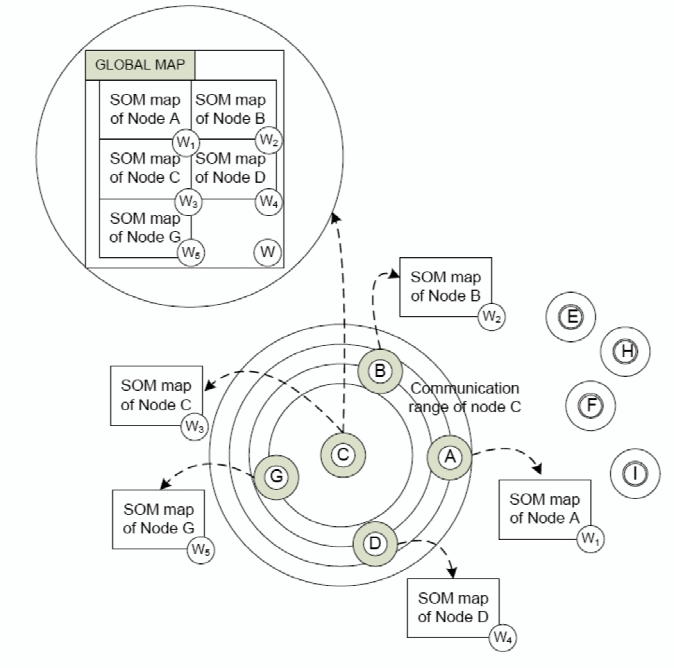

Figure 2: Watermarked emergent self-organized maps of a MANET

in the first layers of the protocol stack. Thus, intrusion detection mechanisms that are based on features selected in the MAC layer are faster regarding the detection delays and the response time. Furthermore, these features make the discrimination between normal and abnormal behavior easier [18].

The proposed intrusion detection model follows a totally distributed architecture. Each node of the MANET should perform its local intrusion detection using local audit data [34]. When the confirmation of other nodes to detect an attack is necessary, local intrusion detectors should cooperate. Furthermore, this cooperation between local intrusion detectors should be held through secure channels.

The IDS architecture we adopt is composed of multiple local IDS agents as illustrated in Figure 1 that are responsible for detecting possible intrusions locally [10]. The collection of all the independent IDS agents forms the IDS system for the MANET. Each local IDS agent is composed of the following components:

Data Collector. It is responsible for selecting local audit data and activity logs.

Intrusion Detection Engine. It is responsible for detecting local anomalies using local audit data. The local anomaly detection is performed using the eSOM classification algorithm. The procedure that is followed in the local detection engine is described below:

- Select labeled audit data and perform the appropriate transformations.
- Compute the classifier using training data and the eSOM algorithm.
- Apply the classifier to test local audit data in order to classify it as normal traffic or attack.
- Perform watermarking in its eSOM map, in order to be sure that it will not be modified and in order to illustrate the security situation and possible existence of intrusions locally in this node.

In Figure 2, nodes A, B, D and G are in the communication range of node C. Each one of nodes A, B, C, D, G creates its own eSOM map and performs watermarking on it (illustrated as W_1, W_2, W_3, W_4, W_5 respectively). Node C selects the local watermarked eSOM maps from its neighbors and creates the global map of its local network. The global map is produced by a concatenation of all the local maps. By observing the global map of its local network, node C is able to have a view of the security status of its neighboring nodes. Based on this information node C selects the appropriate route in order to forward its messages. In order to verify the authenticity and integrity of the global map, node C also performs watermarking on the global map (illustrated as W). Node C, by observing the local maps of all its neighboring nodes and by considering as secure the nodes that are not victims of attacks, performs the selection of the appropriate node for the forwarding of messages.

Thus, each node collects the eSOM maps of its neighbors and uses them in order to have a view about the security of its neighbors, something that can be easily derived by the visual observation of the watermarked (not modified) maps produced by eSOM. After se-

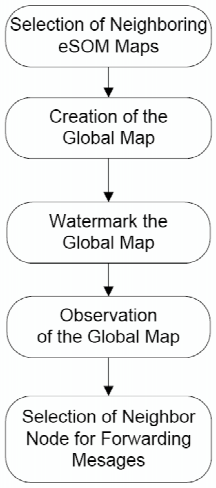

Figure 3: Procedure of selecting the appropriate forwarding node

lecting the local maps from its neighbors, each node creates the global map of the network and performs watermarking on it. Thus, each node is able to know the security status of its local network. The procedure followed is depicted in Figure 3.

Intrusion Response Engine. If the *Intrusion Detection Engine* [21, 22] detects an intrusion then the *Intrusion Response Engine* is activated. The *Intrusion Response Engine* is responsible for sending a local and a global alarm in order to notify the nodes of the mobile ad hoc network about the incident of intrusion. Local alarms are broadcasted to all one hop neighbors of a node and global alarms are sent to all nodes in the transmission range of the attacked node. Moreover, in case that an intrusion is detected though the local eSOM map of a node, the attacked node is not selected to forward information in order to avoid possible loss of information. Special attention should be paid on the function of the *Intrusion Response Engine* in order to avoid possible flooding caused by these intrusion notification messages. Thus, the broadcasted notification of intrusion is restricted to a few hops away from the node where the anomaly has been detected since the neighboring nodes run the greatest risk of possible intrusion.

3 Intrusion Detection Engine Based on Emergent Self-Organizing Maps

Emergent Self-Organizing Maps (eSOM) are based on Kohonen's Self-Organizing Maps (KSOM). KSOM [11] have their base in biology. They belong in the category of unsupervised or competitive learning networks and produce a topological map, which illustrates the input data according to their similarity. The KSOM is trained using only the characteristics of the trained data. The trained KSOMs create clusters of data, where similar vectors of features are located in a specific region in the output space. This is very useful for discovering clusters and relationships in data. The generated mapping is topology preserving. The learning procedure consists of the following steps:

1) Initialise the random weights w_{ij} (also known as codebook vectors of the neurons) with small random values.

2) Use an input pattern x.

3) Calculate the Euclidean distance in Equation (1) [11] between the input data sample x, and each neuron weight w_{ij}. The winner (Best Matching Unit) is chosen as $o(x)$:

$$o(x) = \arg\min_j \|x - w_{ij}\|, j = 1, 2, \ldots, l, \qquad (1)$$

where l is the number of neurons.

4) Adjust all the weights in the neighborhood, in order to achieve the topological mapping, depending on their distance from the winning neuron according to the following equation [15]:

$$\forall j : w_{ij}(t-1) + \alpha(t)\eta(t)(x_i(t) - w_{ij}(t-1),$$

where α is the learning rate, η the neighborhood function and t is the time that was spent in the current context. The neighborhood function η decreases as t increases.

5) Repeat Steps 2, 3, 4, until convergence.

One of the basic disadvantages of KSOM maps is that their abilities are limited to a few neurons. On the other hand, emergent Self-Organizing Maps may expand to some thousands of neurons. A large number of neurons in an eSOM is necessary in order to achieve emergence. The cooperation of such a big number of neurons leads to structures of a higher level. The clustering procedure in eSOMs is performed by observing the whole emergent Self-Organizing Map and not by focusing on its neurons.

There are several techniques to express distance. We have used the distance based (U-Matrix) method in order to visualize the structures generated by eSOM. According to this method [26] the sum (height) of distances between the neuron-weights is represented as the elevation of each neuron. If n is a neuron on the map, $NN(n)$ is the set of one hop neighbors on the map and $w(n)$ is the weight vector associated with neuron n, then the height $U - height(n)$ of each neuron n is given by the following equation [28]:

$$U - height(n) = \sum_{m \in NN(n)} (w(n) - w(m)),$$

where $d(x, y)$ is the distance used in the eSOM algorithm to construct the map. The U-Matrix is a display of the U-heights on top of the grid positions of the neurons on the map. The input data set is displayed and depicted at a 3D landscape. The height will have a large value in areas of the map where one finds a few data points and a small value in areas that represent clusters, creating *hills* and *valleys* respectively.

In our MANET examples, we trained eSOMs with logs of network traffic selected from a simulated MANET (using ns-2) and used the eSOM U-Matrices [28] in order to perform intrusion detection. In our case, a vector represents each log of network traffic with some fixed attributes. Each vector has a unique spatial position in the U-Matrix and the distance between two points defines the dissimilarity of two network traffic logs. The U-Matrix of the trained dataset is divided into *valleys* that represent clusters of normal or attack data and *hills* that represent borders between clusters. Depending on the position of the best match of an input data point that characterizes a connection this point may belong to a *valley* (cluster (normal or attack behavior)) or this data point may not be classified if its best match belongs to a *hill* (boundary). The map that will be created after the training of the eSOM will represent the network traffic. Thus, an input data point may be classified depending on the position of its best match.

Considering the fact that image maps are exposed to the possibility of manipulation, techniques must be applied to eSOM maps in order to verify their authenticity and to detect any modifications of the maps. Watermarking is proposed as such a technique in this paper.

4 Protecting eSOM Maps with Watermarking Techniques

Watermarking techniques have been mainly applied to protect the copyrights of any digital medium by embedding a unique message within the original information [26]. One of the most important requirements of watermarking is the perceptual transparency between the original work and the watermarked one. The watermarked message may have a higher or lower level of perceptibility, meaning that there is a greater or lesser likelihood that a given observer will perceive the difference between the watermarked and the plain image, in our case the eSOM U-Matrix.

We use the *Lattice* and the *Block-Wise* watermarking techniques [26] for the eSOM U-Matrices, which are in the form of images in uncompressed format (bmp). The *Lattice* method has two parameters, the *alpha0* (lattice spacing) and the *beta* (embedding strength), while the *Block-Wise* method has only one parameter, the quantization factor *alpha* to assess the changes. We combined these two watermarking techniques in order to implement a cryptographic encoder-decoder that can be used in order to authenticate the nodes in the MANET.

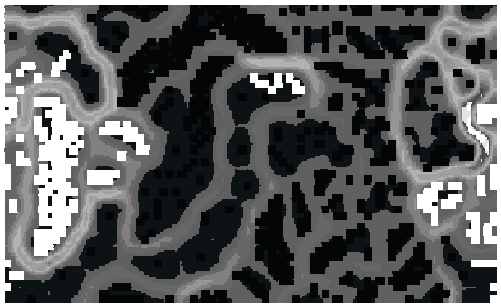

Figure 4: Test image - Emergent SOM U-matrix of a MANET's node

For a fair comparison between the original and the watermarked work there are efficient distortion metrics [9]. Objective criteria are trustworthier in comparison to subjective ones and they are commonly used in the research and development environments. These distortion metrics do not exploit the properties of the *Human Visual System (HVS)* but they provide reliable objective results. There is also an objective criterion that relies on the sensitivity of the eye called the *Watson Perceptual Distance* [33]. This distance is also known as *Just Noticeable Differences (JND)* and consists of a sensitivity function, two masking components based on luminance and contrast masking, and a pooling component. Table 1 gives the metrics that are used more often.

The image we have used to perform our experiments is in bitmap grayscale format with 256×400 resolution. To observe the difference between the original and the watermarked image it is necessary to use the quality measurements of Table 1 [26, 33] and to calculate the ideal values. Supposing that the original and the watermarked test image (Figure 4) are exactly identical, the ideal values (from Table 1) of the test image are presented in Table 2.

In the following paragraphs the *Lattice* and the *Block-Wise* embedding methods are described as well as how the combination is applied to the watermarking of the eSOM U-Matrices.

4.1 Lattice Embedding Method

In a lattice code, each codeword is a point on a regular lattice. The points in a simple N-dimensional lattice can be constructed by adding integer multiples of N distinct vectors. Each message mark w_m is a point in a lattice and is given as the sum of one or more reference marks w_r.

The reference marks are orthogonal to each other. The integer that describes the closest code word to any message vector is calculated by first finding the length of the message vector projected onto the reference mark and then by dividing it by the length and quantizing it to the nearest vector. The lattice watermarking system embeds only one bit per 256 pixels in an image. Each bit is encoded using a trellis code, producing a sequence of four

Table 1: Quality metrics

Mean Square Error (MSE)	The expected value of the square of the error
Signal to Noise Ratio (SNR)	The ratio of a signal power to the noise power corrupting the signal.
Peak Signal to Noise Ration (PSNR)	The maximum Signal to Noise Ratio
Image Fidelity (IF)	The process of rending an image accurately without any visible distortion of information loss.
Normalized Cross Correlation (NCC)	Measure of similarity of two signals.
Correlation Quality (CQ)	The deviation of two messages.
Watson Distance (WD)	The points or pixels distance between two images.

bits. Trellis coding is a convolutional code, the number of states in the code is $2^3 = 8$ and the possible outputs are $2^4 = 16$. So after the trellis coding procedure, the bits have to be embedded in 256 pixels. This means that each of the four bits is embedded in $256/4 = 64$ pixels. The image is divided in blocks of 8×8 pixels in order to host the bits. The reference pattern consists of 8×8 random pixels. The pixel values are normalized to have zero mean and unit variance. Each bit is embedded by correlating a block against the 8×8 reference pattern, and by quantizing the result to an odd or even integer. The reference pattern, that is added to the 8×8 block according to the index of the closest point in the sublattice ($z_m[i]$), is computed by the following formulas:

$$l[i] = \frac{c_i * w_r}{|w_r|},$$

where c_i is the i^{th} block of the image, w_r is the reference pattern and $l[i]$ is the length of the c_i projected onto w_r

$$z_m[i] = 2 \left\lfloor \frac{l[i]/(\beta|w_r) - m_c[i]}{2} + 0.5 \right\rfloor,$$

where $m_c[i]$ is the corresponding message and the added pattern $w_{\alpha_0 i}$ is given by:

$$w_{\alpha_0 i} = \alpha_0(\beta z_m[i] w_r - c_i).$$

The parameters in the embedding process are: α_0 (alpha0) that represents the embedding strength and β (beta) that represents the lattice spacing. At the decoder side, $z[i]$ is first computed by Equation(2) and then the least significant bit of it is detected. The coded message is then decoded with the trellis decoder:

$$z[i] = \left\lfloor \frac{c_i w_r}{\beta w_r * w_r} + 0.5 \right\rfloor \quad (2)$$

4.2 Block-Wise Embedding Method

The *Block-Wise* embedding method involves the basic properties of the JPEG compression where the *Discrete Cosine Transform (DCT)* domain takes place. Both the encoder and the decoder use these properties in order to achieve the embedding and the extraction processes respectively. The predefined parameters is a strength parameter alpha (α), which is used as the scaling factor of the luminance quantization matrix.

Table 2: Ideal values of the test image

Quality Measurements	Ideal Values
MSE	0
SNR (dB)	94
PSNR (dB)	110
IF	100
NC	1
CQ	138.178
Watson Distance	0

Four bits are embedded in the high-frequency *DCT* of each 8×8 (64 pixels) block in the image. In the *Lattice* method, one bit per 256 pixels is embedded. It seems that by using the *Block-Wise* method the image can host 16 times more information. As it was mentioned before the embedding takes place in the high-frequency *DCT* coefficients and not in the low-frequency ones in order to avoid any visual differences that would lead to unacceptably poor fidelity. More precisely, we have used 28 coefficients which means that each bit is embedded in seven coefficients.

The seven coefficients that are going to host one bit are chosen randomly according to a seed number in Equation (3). Thus, each coefficient is involved in only one bit. The next step is to divide each coefficient by its corresponding quantization factor and to round to the nearest integer, i.e.

$$C_I[i] = \left\lfloor \frac{C[i]}{\alpha q[i]} + 0.5 \right\rfloor, \quad (3)$$

where $q[i]$ is the corresponding value of the luminance matrix.

Then the algorithm takes the least significant bit of the resulting seven $C_I[i]$ integers and exclusive-ors (XOR) them to obtain a bit value b_e. The bit value, which has to be embedded, is b. When $b_e \neq b$ one of the seven integers $C_I[i]$ is randomly chosen, depending on which one will cause the least fidelity impact. Let $C_{wI}[i]$ denote the result. That is $C_{wI}[i] = C_I[i]$ for all I in case of $b_e = b$, unless $b_e \neq b$, in which case the least significant bit of one member of the seven $C_{wI}[i]$ is multiplied by the corresponding quantization factors to obtain the watermarked versions of DCT coefficients. Then the equation for the result is given by:

$$C_w[i] = \alpha q[i] C_{wI}[i].$$

At the decoder the procedure is exactly the same. From each 8×8 block the least significant bit b_e is extracted from each of the seven coefficients and it is compared with the embedded one b. If the two bits are different, the corresponding block is not authenticated and it is marked as corrupted.

4.3 Combined Method

The *Lattice* algorithm uses error control coding. Its functionality is based on constructing orthogonal reference marks to be used in the embedding process. But in case that somebody modifies a number of blocks, the decoder will not detect it since it uses trellis coding. It is obvious that, if a continuous number of blocks has been changed, the decoder will not be able to extract the correct sequence of bits. The *Lattice* method embeds one bit per 256 pixels and the quality of the watermarked image is very high. On the other hand, the *Block-Wise* method embeds four bits per 64 pixels. The payload that can be hosted is larger compared to the *Lattice*, a fact which is very useful in low-resolution images. But the quality of the produced image is not as good, since the user can exploit the absence of error control. Any modification of the watermarked image can be located by comparing the extracted message with the original. Questions of who and why modified the image can be answered easily. Thus, in cases where the quality and the ability to notice the corrupted blocks have the same importance, it is essential to combine the two embedding methods.

The combination of the two embedding methods can be implemented in a cryptographic encoder-decoder. A node can give a message like its ID and a short description (i.e. the number of one-hop neighbors). Then a unique description of the image can be used (i.e. the sum of the pixel values of the four blocks in the corners). These three messages are inserted in a hash function and then the value is encrypted with a 1024-bit secret key. The signature with the short and the extended description are embedded with the *Lattice* method while the message is embedded with the *Block-Wise* algorithm. The design of the encoder is illustrated in Figure 5.

From the watermarked version of the image, at the decoder's side, the signature, the short description and

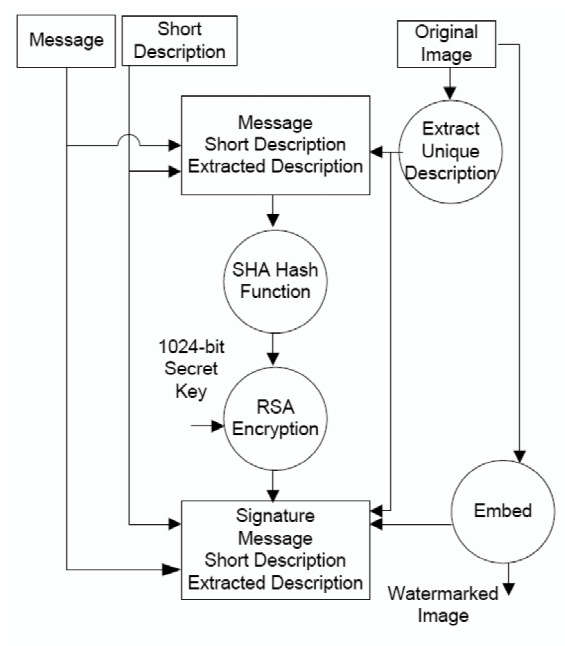

Figure 5: Cryptographic encoder

the unique description are extracted with the *Lattice* method, while the message is extracted with the *Block-Wise* method. The unique description is evaluated again and it is copared with the extracted one. So the first step is to verify if the unique descriptions match. In case of copying the watermark and embedding it in another image, the extracted description will not be the same. Because the pixel values of the image have been slightly changed to host the watermark, the extracted description cannot be exactly the same, but only very close. Therefore, some upper and lower boundaries have been determined based on the ideal values of the test image (see Table 2). The next step is to decrypt the signature using the 1024-bit public key and get the hash value. The message, the short description and the unique description that have been extracted, are used as an input to the hash function. The obtained hash value is then compared to the one decrypted from the signature. The second step of the decoder is to verify if the decrypted hash value matches exactly the one calculated at the decoder. If both the stages of the hash values and the unique descriptions are valid, the authentication process is successful. The design of the decoder is presented in Figure 6.

5 Performance Evaluation

5.1 Evaluation of the Detection Engine

To evaluate the feasibility of our intrusion detection engine we have conducted a series of experiments. For our experiments we have made the assumption that the network has no preexisting infrastructure and that the em-

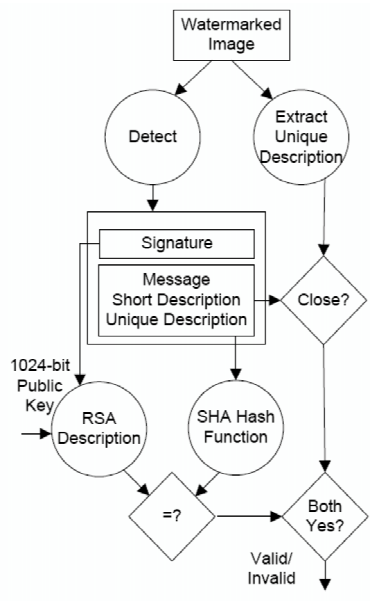

Figure 6: Cryptographic decoder

ployed ad hoc routing protocol is AODV.

We implemented the simulator within the ns-2 library. Our simulation modeled a network of 50 hosts placed randomly within an 1800×1000 m^2 area. Each node has a radio propagation range of 250 meters and a channel capacity of 2 Mbps. The nodes in the simulation move according to the "random way point" model. At the start of the simulation, each node waits for a pause time and then it randomly selects a destination and it moves towards that with a speed uniformly lying between zero and the maximum speed. On reaching this destination it pauses again and repeats the above procedure till the end of the simulation. The minimum and maximum speed is set to 0 and 10 m/s, respectively, and the pause times are set at 0, 20, 50, 70 and 200 sec. A pause time of 0 sec corresponds to the continuous motion of the node and a pause time of 200 sec corresponds to a stationary node.

We evaluated the performance of our proposed intrusion detection module for 5, 10, 15 and 20 malicious nodes. In each case the number of all nodes in the network is set to 50. The malicious behavior is carried between the 50th and 200th sec. The nodes perform normally between 0 and 50 sec. These parameters result in a network with a rather high mobility and a high traffic activity.

On average, twenty traffic generators were developed to simulate a TCP data rate to ten destination nodes. This traffic pattern results in twenty connections among source and destination nodes. The sending packets have random sizes and exponential inter-arrival times. The sources and the destinations are randomly selected with uniform probabilities. The mean size of the data payload is 512 bytes. Each run is executed for 200 sec of simulation time with a feature-sampling interval of one sec. We used the IEEE 802.11 Distributed Coordination Function (DCF) as the medium access control protocol. The mobility of the nodes is random determined by scenario files that are generated by the scene generator of ns-2. A free space propagation model with a threshold cutoff was used in our experiments. In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, i.e., radio capture.

In the experiments we simulated a constant selective packet-dropping attack where the attacker simply discards all data packets while it functions legitimately concerning routing and MAC layer packets. This type of attack is extremely difficult to detect if we consider that packet dropping can happen due to a malicious behavior or due to mobility. To add to the problem the malicious node may exhibit this malicious behavior when it is most advantageous to him and not from the beginning.

The statistical features we have used have been introduced by Liu et al. [18] in their proposed approach for performing intrusion detection in the MAC layer. These features are as follows:

- *Network allocation vector (NAV):* it is a node specific characteristic, which depicts the time that the node will occupy the medium for sending its messages.
- *Transmission traffic rate:* it indicates the rate of the transmitted packets.
- *Reception traffic rate:* it indicates the rate of the received packets.
- *Retransmission rates of RTS packets:* it indicates the rate of the Ready-To-Send packets that are retransmitted by the monitoring node. A high value of this feature suggests a possible packet dropping attack.
- *Retransmission rates of data packets:* it indicates the rate of the data packets that are retransmitted by the monitoring node. A high value of this feature suggests a possible packet dropping attack.
- *Active neighbor node count:* it represents the number of neighbor nodes that have data transmission activities.
- *Forwarding node count:* it represents the number of neighbor nodes that communicate directly with the monitoring node.

In order to avoid having the attributes of some input vectors disproportionally influence the results, it is necessary to normalize the input data. Many methods have been used in the literature for the data normalization. We have normalized the data with mean zero and variance one, a technique that produces very good results in most cases as reported in the literature. For the experimental results we have used the Databionics eSOM tool [29, 30].

In order to perform clustering with eSOM U-Matrices we followed the preceding procedure. The best matches of the trained dataset and, thus, the corresponding dataset

are manually grouped into clusters representing normal and attack behavior. Thus, we identify the regions of the map that represent a cluster that can be used for the classification on new datasets. The eSOM of a trained dataset is depicted in Figure 4. As it can be clearly seen the training data set has been divided in two classes that are very well distinguished, the normal data class (dark color) and the packet dropping data class (light color). In order to make sure that our intrusion detection engine will always provide efficient and accurate results we should update our trained eSOM U-matrix according to the new conditions concerning mobility.

The detection rate for the all cases examined in over 80% while the false alarm ranges around 20%. The high level of false alarms is mainly caused because of the difficulty that the classifier (eSOM) faces to discriminate the change in the behavior of a node (depicted in the selected features) caused either due to mobility reasons or due to malicious behavior. A more detailed presentation of these results is presented in our paper [20].

Our intrusion detection engine presents a rather high detection rate with the advantage of the visual representation of normal-attack state in a MANET. Moreover, the intrusion detection engine has the ability to immediately respond in the case of a likely intrusion by selecting the more secure node as indicated by its U-Matrix map for forwarding the information. In order to verify the reliability and possible alteration of the maps there is a need of watermarking.

5.2 Evaluation of Watermarking Procedure

In order to evaluate the performance and the efficiency of the embedding methods, a large number of tests were performed. Several cases were considered, each with a different variable parameter. First, the impact of the *Lattice* embedding method on the image quality is presented. Then, the results using the *Block-Wise* embedding method are illustrated and finally the section will conclude with the observations using a combination of the embedding methods.

5.2.1 Lattice Embedding Method

In the *Lattice* method the maximum number of the embedded bits can be 400 (one bit per 256 pixels). The quality metrics that are used to evaluate the differences between two images have been presented in Table 1. The tests were executed for a range of the parameter's values in order to conclude in the best values. The parameters are the embedding strength beta (β) and the lattice spacing alpha0 (α_0). The range of α_0 was from 0.35 to 5.33 and the range of β from 0.7 to 1.1. The increase steps for α_0 was 0.02 and for β 0.1. The measurement values for the *Lattice* method are very close to the ideal ones. More specifically, the direction towards zero is achieved using low values of α_0 in case of MSE. If at the same time the value of β that is used, is low the MSE is decreased even further. In the case of SNR and PSNR, the result values are higher when the parameters α_0 and β are low. The Image Fidelity (IF) is defined as a percentage of how identical the images are. So the value of 100% is considered to be the optimum and as can be noticed from the graphs, the results are very close to this. Concerning the NC and CQ quality measurements, it is observed that their measurements are closer to the ideal ones, as the values of α_0 and β decrease. Finally, all the above observations are also justified from the *Watson distance* metric which is based on luminance, contrast, and pooling masking.

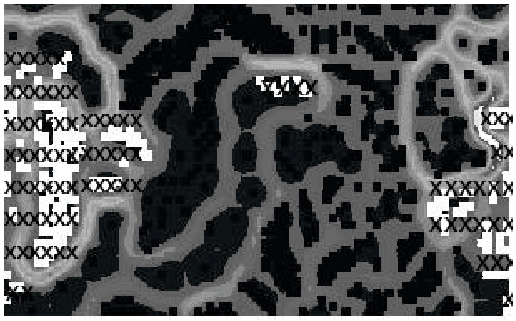

Figure 7: Marked image for the test of the cryptographic encoder-decoder

Therefore, someone could suggest that the optimum parameter values are those that give the best results. These values could be even equal to zero, but at the decoder's side not all the bits are extracted right. Specifically using low values of α_0 and β the decoder is not able to get the right embedded bits. In conclusion, it can be said that a trade-off between the quality results and the decoder's result is necessary in order to determine the optimum values. From the tests we concluded that suggested values could be $\alpha_0 \approx 0.8$ and $\beta = 0.9$. In Table 3 some evaluated values of the experiments are given in order to justify all the above observations. The watermarked version of the test image presents no noticeable differences from the test image (Figure 7).

5.2.2 Block-Wise Embedding Method

In the case of the *Block-Wise* method, the tests were executed for the same image in order to be comparable with those for the *Lattice* method. One major difference is the number of bits that are embedded. Since the method embeds four bits in every 64 pixels and the image has 102,400 pixels in total, the number of bits that can be hosted is 6406. The size of the information that can be watermarked is significantly higher, 16 times more than the size for the *Lattice* method. Thus, before even executing the test it is expected that the results will not be as good, especially for the values of the quality metrics. The only parameter in the *Block-Wise* method is the one responsible for the quantization of the luminance matrix and is called alpha (α).

Table 3: Result values of the lattice method

Lattice	MSE	SNR	PSNR	IF	NC	CQ	Watson	Right Bits
$\alpha_0 = 0.35$, $\beta = 1$	0.019	64.49	70.21	100	1	137.04	8.144	370
$\alpha_0 = 1.01$, $\beta = 0.9$	0.27	51.84	56.72	99.996	1	136.97	21.178	400
$\alpha_0 = 1.85$, $\beta = 0.8$	0.97	49.97	53.12	99.993	1	136.97	51.687	400

The observation of the results proves what has been stated in the beginning. The values of the quality metrics are not as good in comparison to those from the *Lattice* method. The value of the MSE is higher than the ideal zero value. The values of the SNR and PSNR, which are used widely as performance parameters, show that as the value of β increases the results become worse. In the case of the IF, NC, CQ, the measurements seem to be far from the ideal values as α takes higher values. The same conclusion can be derived for the perceptual distance given from the Watson model, where the results are worse as the value of α increases. Some values of the quality measurements are given in Table 4.

According to the above, it seems that as the value of α increases, the watermarked image has poorer fidelity. So the optimum value of the parameter could be possibly a small one i.e. 0.01. But it seems that values below 0.01 do no allow the decoder to get the right message. The chosen value of α depends on how sensitive the method the user wants it to be in order to locate the corrupted bits and mark the corresponding blocks. Higher values increase the sensitivity but at the same time the quality of the image reduces. Thus, it is again necessary to make a trade-off between the results and the sensitivity. A possible suggested value could be $\alpha \approx 0.2$. The watermarked version of the original image produced with the *Block-Wise* method has no visible difference from the test image (Figure 4).

5.2.3 Combined Method

In order to perform watermarking in the eSOM U-Matrices we exploit the advantages of the above presented embedding methods, the *Lattice* and the *Block-Wise* method. The *Lattice* algorithm provides high quality for the watermarked image but the number of bits that are embedded is only one bit per 256 pixels. On the other hand, the *Block-Wise* method embeds four bits per 64 pixels but with the cost of poor quality of the produced image. Furthermore, the absence of error control in the *Block-Wise* method gives us the advantage of being able to easily locate any alterations of the watermarked image. In eSOM U-Matrices the part that is likely to be illegally altered is watermarked with the *Block-Wise* method, while the rest of the image is watermarked with the *Lattice* method. Thus, in the eSOM U-Matrix (Figure 4) the areas with light color, representing the attack data class, i.e. the packet dropping data class will be watermarked using the *Block-Wise* method while the rest of the eSOM U-Matrix (the normal data class (dark color)) will be watermarked using the *Lattice* method. This watermarking gives us the ability to have a high quality image and at the same time if an adversary changes, for example, the area of attack data class the combined algorithm will be able to determine the modified pixels. This can be achieved by comparing the extracted message with the original one.

The message is embedded in the part of the image that is watermarked with the *Block-Wise* method, while the signature, the short and the extracted description are embedded in the large part of the image. Since the *Lattice* method gives better results than the *Block-Wise*, it was expected that the produced result values would be in between the values of those produced by the two methods. Indeed, the results were not as good as those of the *Lattice's* but at the same time they were better than those of the *Block-Wise's*. In Table 5 some results of the combination of the two methods are given in order to compare them to those of the two methods when they are used individually. The presented results justify that the combination produces quality measurements between the two methods. In Table 6 are presented the maximum number of bits that can be hosted in the image using the two embedding methods and a combination of them.

In order to verify that a possible modification of the eSOM U-Matrix can be detected by the decoder, we performed an additional test using the eSOM U-Matrix of Figure 4. In the watermarked version, the light area representing the existence of attack in a node of the MANET was changed and this image was inserted to the decoder in order to verify its authenticity. The decoder determined the modification and informed us that the modification has failed. By observing Figure 7 it is clear that the decoder has successfully located the modified blocks. Therefore, the whole implementation of the cryptographic encoder-decoder performs as expected.

To ensure the applicability of the proposed approach in the MAC layer and in real ad hoc environments with resource constraints, all the necessary computations of the watermarking technique should be pre-calculated. Thus, the only computational complexity derives from the generation and verification of the digital signature. Further-

Table 4: Result values of the block-wise method

Block-Wise	MSE	SNR	PSNR	IF	NC	CQ	Watson	Right Bits
$\alpha_0 = 0.03$	0.312	44.11	62.18	99.9981	0.99997	138.9	12.144	6012
$\alpha_0 = 0.16$	4.324	36.22	52.32	99.9701	0.99988	137.902	108.972	6406
$\alpha_0 = 0.33$	11.321	31.45	44.29	99.8926	0.99978	137.123	309.456	6406

Table 6: Maximun number of embedded bits

	Lattice	**Block-Wise**	**Combined**
Max Embedded Bits	400	6406	$\geqslant 4100$

Table 5: Result values of the combined embedding methods

$\alpha_0 = 0.93$, $\beta = 1$, $\alpha = 0.1$	Lattice α_0, β	Block-Wise α	Combined, α_0, β, α
MSE	0.385	1.785	0.394
SNR	44.2	40.45	45.74
PSNR	53.14	47.25	51.98
IF	99.9972	99.9978	99.9975
NC	0.99999	0.99902	0.99998
CQ	139.457	139.578	139.457
Watson-Distance	31.415	59.788	31.499
$\alpha_0 = 1.53$, $\beta = 0.8$, $\alpha = 0.2$	Lattice α_0, β	Block-Wise α	Combined, α_0, β, α
MSE	0.557	4.121	0.74
SNR	44.08	32.97	42.41
PSNR	51.14	40.54	49.75
IF	99.9968	99.9482	99.9836
NC	0.99998	0.99989	0.99997
CQ	139.784	139.78	139.785
Watson-Distance	49.145	155.518	50.002

more, the overhead of the proposed watermarking approach is the same with the key length that we are using in the signature algorithms, i.e. 1024-bits.

6 Conclusions and Future Work

In this paper, we have presented an intrusion detection engine that is part of a local IDS agent in every node of a MANET. The collaboration of all the local IDS agents composes an IDS for MANET. The proposed intrusion detection engine is based on emergent SOMs a special and efficient class of neural networks that generates as an output a map and provides visual representation of the classification performed. We exploited the advantage of visualizing the network traffic and examined how eSOM performs in classifying normal and attack behavior in MANET based on MAC layer features and we exploited the advantage of visualizing network traffic. Using eSOM each node of the MANET creates its local eSOM map as well as the global map of its neighbors. The local and global eSOM maps provide us the important advantage of being able to have a visual representation of the security status of each MANET node. Thus, each node has the option to select a secure routing path for packet forwarding by avoiding compromised neighbors.

For the authentication of the local and the global maps an innovative and efficient watermarking method is proposed which derives from the combination of two watermarking embedding methods, the *Lattice* and the *Block-Wise*. The proposed watermarking method exploits the advantages of the *Lattice* and the *Block-Wise* method in order to produce the most efficient and reliable results. The most sensitive part of the eSOM map that represents the existence of an attack in a node being the most sensitive part of the map is watermarked with the *Block-Wise* method and the rest of the map with the *Lattice* embedding method.

We exploit the significant advantages of visual representation and watermarking in MANET, two research areas that have not previously used in the research field of MANET. Special attention should be paid to the fact that the detection engine could be employed in various routing protocols. We plan to select features from other layers (e.g. network layer) in order to examine the performance of the proposed approach for the detection of other type of attacks.

Acknowledgements

We would like to thank the reviewers for providing valuable feedback and Christos Dimitrakakis for additional proofreading. This work was partially supported by the Netherlands Organization for Scientific Research (NWO)

under the RUBICON "Intrusion Detection in Ubiquitous Computing Technologies" grant awarded to Aikaterini Mitrokotsa.

References

[1] F. Anjum, D. Subhadrabandhu, and S. Sarkar, "Signature-based Intrusion detection for wireless ad hoc networks: A comparative study for various routing protocols," *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC' 03)*, vol. 3, pp. 2152-2156, Orlando, Florida, Oct. 2003.

[2] T. Bhaskar, N. Kamath B, and S. D. Moitra, "A hybrid model for network security systems: Integrating intrusion detection system with survivability," *International Journal of Network Security*, vol. 7, no. 2, pp. 249-260, 2008.

[3] M. S. Bouassida, I. Chrisment, and O. Festor, "Group key management in MANETs," *International Journal of Network Security*, vol. 6, no. 1, pp. 67-79, 2008.

[4] C. W. Chen, M. C. Chuang, and C. S. Tsai, "An efficient authentication scheme between MANET and WLAN on IPv6 based Internet," *International Journal of Network Security*, vol. 1, no. 1, pp. 14-23, 2005.

[5] T. M. Chen, and V. Venkataramanan, "Dempster-shafer theory for intrusion detection in ad hoc networks," *IEEE Internet Computing*, vol. 9, no. 6, pp. 35-41, Nov. 2005.

[6] Databionic ESOM Tools, 2008. (http://databionic-esom.sourceforge.net/devel.html)

[7] H. Deng, Q. Zeng, and D. P. Agrawal, "SVM-based intrusion detection system for wireless ad hoc networks," *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC' 03)*, vol. 3, pp. 2147-2151, Orlando, Florida, Oct. 2003.

[8] M. Eid, H. Artail , A. Kayssi, and A. Chehab, "LAMAIDS: A lightweight adaptive mobile agent-based intrusion detection system," *International Journal of Network Security*, vol. 6, no. 2, 2008, pp. 145-157

[9] T. Furon, "A Survey of watermarking security," *Proceedings of the 4th International Workshop on Digital Watermarking*, pp. 201-215, Siena, Italy, Sep. 2005.

[10] L. Girardin, "An eye on network intruder-administrator shootouts," *Proceedings of the 16th international conference on Information Security 2001 france, Proceedings of the Workshop on Intrusion Detection and Network Monitoring, USENIX Association*, pp. 19-28, Santa Clara, California, USA, Apr. 1999.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, 2nd Edition, New Jersey, USA, 1999.

[12] Y. Huang, W. Fan, W. Lee, and P. Yu, "Cross-feature analysis for detecting ad-hoc routing anomalies," *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems*, pp. 478-487, Rhode Island, Greece, May 2003.

[13] Y. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, pp. 135-147, Fairfax, Virginia, USA, Oct. 2003.

[14] O. Kachirski and R. Guha, "Intrusion detection using mobile agents in wireless ad hoc networks," *Proceedings of the IEEE International Workshop on Knowledge Media Networking (KMN' 02)*, pp. 153-158, Kyoto, Japan, July 2002.

[15] N. Komninos, D. Vergados, and C. Douligeris, "Detecting unauthorized and compromised nodes in mobile ad hoc networks," *Journal in Ad Hoc Networks*, vol.5, no.3, pp. 289-298, Elsevier Press, Apr. 2003.

[16] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data: A state-of-the-art overview," *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20-46, Sep. 2000.

[17] C. T. Li and Y. P. Chu, "Cryptanalysis of threshold password authentication against guessing attacks in ad hoc networks," *nternational Journal of Network Security*, vol. 8, no. 2, pp. 166-168, 2009.

[18] Y. Liu, Y. Li, and H. Man, "MAC layer anomaly detection in ad hoc networks," *Proceedings of 6th Annual IEEE SMC Information Assurance Workshop (IAW' 05)*, pp. 402-409, June 2005.

[19] S. Makki, N. Pissinou, and H. Huang, "The security issues in the ad hoc on demand distance vector routing protocol (AODV)," *Proceedings of the 2004 International Conference on Security and Management*, pp. 427-432, Las Vegas, Nevada, USA, June 2004.

[20] A. Mitrokotsa, N. Komninos, and C. Douligeris, "Intrusion detection and response in ad hoc networks," *Advances in Ad Hoc Network Security, International Journal on Computer Research*, vol. 15, no. 1, Nova Science Publishing Inc., 2007.

[21] A. Mitrokotsa, N. Komninos, and C. Douligeris, "Intrusion detection with neural networks and watermarking techniques for MANET," *Proceedings of IEEE International Conference on Pervasive Services (ICPS' 07)*, pp. 118-127, Istanbul, Turkey, July 2007.

[22] A. Mitrokotsa, N. Komninos, and C. Douligeris, "Towards an effective intrusion response engine combined with intrusion detection in ad hoc networks," *Proceedings of the Sixth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net' 07)*, pp. 137-144, Corfu, Greece, June 2007.

[23] R. Páez, C. Satizbal, and J. Forn, "Cooperative itinerant agents (CIA): Security scheme for intrusion detection systems," *Proceedings of the International Conference on Internet Surveillance & Protection (ICISP' 06)*, pp. 26, Cote d'Azur, France, Aug. 2006.

[24] A. Patcha, and J. M. Park, "A game theoretic formulation for intrusion detection in mobile ad hoc networks," *International Journal of Network Security*, vol. 2, no. 2, pp. 131-137, 2006.
[25] A. Rawat, P. D. Vyavahare, and A. K. Ramani, "Enhanced DSR for MANET with improved secured route discovery and QoS," *International Journal of Network Security*, vol. 5, no. 2, pp. 158-166, 2007.
[26] J. Seitz, *Digital Watermarking for Digital Media*, Information Science Publishing, ISBN: 1591405181, 2005.
[27] C. Y. Tseng, P. Balasubramanyan, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN' 03)*, pp. 125-134, Fairfax, Virginia, USA, Oct. 2003.
[28] A. Ultsch, "Data mining and knowledge discovery with emergent SOFMs for multivariate time series," *Kohonen Maps*, pp. 33-46, 1999.
[29] A. Ultsch, "Maps for visualization of high-dimensional data spaces," *Proceedings of Workshop on Self-Organizing Maps (WSOM' 03)*, pp. 225-230, Kyushu, Japan, Sep. 2003.
[30] A. Ultsch and F. Moerchen, *ESOM-maps: Tools for Clustering, Visualization, and Classification with Emergent SOM*, Technology Report, Depatrment of Mathematics and Computer Science, University of Marburg, Germany, 46, 2005.
[31] X. Wang, D. S. Reeves, S. F. Wu, and J. Yuill, "Sleepy watermark tracing: An active network-based intrusion response framework," *Proceedings of the 16th International Conference of Information Security (IFIP/SEC' 01)*, pp. 369–384, Paris, France, June 2001.
[32] S. Xiang and J. Huang, "Analysis of quantization-based audio watermarking to D/A and A/D conversions," *International Journal of Network Security*, vol. 3, no. 3, pp. 230-238, 2006.
[33] Q. Zhang, *New Techniques for Digital Watermarking*, ProQuest / UMI, ISBN: 0542283778, Dec. 2006.
[34] Y. Zhang, W. Lee, and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Wireless Networks*, vol. 9, pp. 545-556, 2003.

Aikaterini Mitrokotsa is a postdoctoral researcher at the Faculty of Electrical Engineering, Mathematics and Computer Science of Delft University of Technology in the Netherlands. Formerly, she held a position as a visitor assistant professor in the Department of Computer Science at the Free University (Vrije Universiteit) in Amsterdam. In 2007, she received a Ph.D in Computer Science from the University of Piraeus. Dr. Mitrokotsa's main research interests lie in the area of network security, intrusion detection systems, denial of service attacks, neurocomputing and machine learning applications to RFID, wired, wireless ad hoc and sensor networks security. She has been active both in European and National research projects while recently she has been awarded the Rubicon Research Grant by the Netherlands Organization for Scientific Research (NWO).

Nikos Komninos received his B.Sc. degree in Computer Science & Engineering from the American University of Athens, Greece in 1998, his M.Sc. degree in Computer Communications & Networks from Leeds Metropolitan University, UK in 1999 and his Ph.D. degree in Communications Systems from Lancaster University, UK in 2003. Dr. Komninos is currently an Assistant Professor in Applied Cryptography and Network Security, member of the Algorithms and Security Group at Athens Information Technology (affiliate of Carnegie Mellon University) and Instructor of the Applied Cryptography, ICT Security and Cryptography & Computer Security postgraduate courses. His current research areas of interest include authentication, key agreement and intrusion detection in ad hoc networks, design and evaluation of efficient encryption algorithms, attack analysis of cryptographic protocols, transport / network layer security, smart cards and biometrics security. He has been invited as technical program committee, guest editor in international conferences and journals and speaker with honours in the field. Dr. Komninos is also senior member in various international societies (IEEE, ACM).

Christos Douligeris received the Diploma in Electrical Engineering from the National Technical University of Athens in 1984 and the M.S., M.Phil. and Ph.D. degrees from Columbia University in 1985, 1987, 1990, respectively. He has held positions with the Department of Electrical and Computer Engineering at the University of Miami, where he reached the rank of associate professor and was the associate director for engineering of the Ocean Pollution Research Center. He is currently a professor at the Department of Informatics of the University of Piraeus, Greece. He has served in technical program committees of several conferences. His main technical interests lie in the areas of performance evaluation of high speed networks, neurocomputing in networking, resource allocation in wireless networks and information management, risk assessment and evaluation for emergency response operations. He was guest editor of a special issue of the IEEE Communications Magazine on security for Telecommunication Networks and he has prepared a book on Network Security by IEEE Press/Wiley. He is an editor of the IEEE Communications Letters, a technical editor of IEEE Network, Computer Networks (Elsevier), International Journal of Wireless and Mobile Computing (IJWMC) and the Euro Mediterranean Journal of Business (EMJB).