Intrusion Detection with Neural Networks and Watermarking Techniques for MANET

Aikaterini Mitrokotsa[*], Nikos Komninos[**] and Christos Douligeris[*]
[*] Department of Informatics, University of Piraeus,
80 Karaoli & Dimitriou Str. Piraeus, 18534, Greece
[**] Athens Information Technology,
19002 Peania Attiki, Greece
{mitrokat,cdoulig}@unipi.gr
nkom@ait.edu.gr

***Abstract*— In mobile ad hoc networks (MANET) specific Intrusion Detection Systems (IDSs) are needed to safeguard them since traditional intrusion prevention techniques are not sufficient in the protection of MANET. In this paper we present an intrusion detection engine based on neural networks combined with a protection method, which is based on watermarking techniques. We exploit the advantages of information visualization and machine learning techniques in order to achieve intrusion detection. Then, we authenticate the maps produced by the application of the intelligent techniques using a novel combined watermarking embedded method. The performance of the proposed model is evaluated under different traffic conditions, mobility patterns and visualization metrics, showing its high efficiency.**

I. Introduction

Mobile ad hoc networks (MANET), also called spontaneous networks, are comprised of a collection of dynamic cooperating peers and consist one of the most promising wireless technologies. The peer nodes in a MANET may show a short duration in their membership with many joins and leaves from the network. They may also employ a multi-hop information transfer without relying on an a-priori infrastructure. The mobile devices in a MANET create a wireless communication channel whereas, each of them contributes in the routing decisions of the network since there are no central stations. Mobile nodes communicate directly with nodes in their vicinity and they relay messages on behalf of others in order to enable communication with devices not in direct radio-range of each other.

The main advantages that MANET present are flexibility, adaptability, easy collaboration and efficient communication in infrastructure-less environments. Because of the special advantages that wireless ad hoc networks present, their applications vary from battlefield scenarios to recovery operations in case of disasters, such as in hurricanes, floods and terrorist acts.

Although MANET present many advantages, they also present a number of inherent vulnerabilities that increase their security risks ([1], [2]). MANET are often subject to eavesdropping, signal jamming and other types of attacks, due to the open medium, the dynamically changing topology, the lack of a centralized monitoring and management point, the limited resources and the lack of physical security of the member nodes.

Intrusion Detection is an indispensable second line of defense since traditional prevention mechanisms are not strong enough to protect MANET. In this paper, we present an Intrusion Detection engine that is part of a local Intrusion Detection System (IDS) composed of a Data Collector, an Intrusion Detection engine and an Intrusion Response engine. We focus on the design of the Intrusion Detection engine that is based on a type of neural networks known as emergent Self-Organizing Maps (eSOMs). By combining machine learning, information visualization and watermarking techniques, we are able to evaluate how secure a MANET is against attacks.

In particular, each ad hoc node of the MANET creates a map that illustrates 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 in order to perform secure and efficient routing by avoiding paths that include nodes which are victims of attacks. Watermarking techniques are then applied in order to prevent the possible modification of the produced maps. The proposed intrusion detection approach uses a combined watermarking technique that derives from Lattice and Block-Wise ([3], [4], [5]) methods.

The success of this method is based on the exploitation of the main advantages of neural networks and watermarking techniques in the design of the Intrusion Detection engine, which is part of a local IDS agent. We exploit the great advantage of tolerance towards imprecise data of neural networks in order to classify normal against abnormal behavior in MANET. eSOMs provide a visual representation of the normal-attack state in each node of the network. Hence, each node can determine whether a neighboring node is under attack and it can subsequently forward its messages accordingly. Furthermore, with watermarking techniques the nodes of the MANET can be authenticated and the integrity of the maps produced by eSOM can be securely verified.

The proposed approach presents some unique characteristics that have not been used in the area of MANET before. We propose an Intrusion Detection approach based on neural networks which exploits the advantages of information visualization in order to achieve direct response in case of possible intrusions. Furthermore, we use watermarking techniques in order to ensure that the proposed derived maps that employ information visualization will not be altered by possible malicious attackers.

This paper is organized as follows. Section 2 presents related work of intrusion detection approaches in mobile ad hoc networks as well as the use of watermarking techniques in information security. Section 3 discusses the intrusion detection model this paper is based on while section 4 presents a functional description of the proposed intrusion detection engine and the underlying classification algorithm used. Section 5 presents the watermarking technique as it is applied in the authentication of the maps produced by eSOMs. In section 6 the performance evaluation of the intrusion detection engine and the proposed watermarking technique are presented. Section 7 concludes the paper and discusses some future work.

II. Related Work

A. *Intrusion Detection in Mobile Ad Hoc Networks*

The lack of centralized choke points makes the monitoring of network traffic in MANET extremely difficult. Thus, intrusion detection mechanisms developed for wired networks can not be easily adopted in wireless ad hoc networks due to the stringent requirements these networks present. But even if these concentration points existed, their locations would continuously change due to mobility, making the deployment of a distributed intrusion detection approach in MANET a necessity. Additionally, the frequent lost and reset connections make the discrimination between a new qualified operation after a disconnection and an intrusion even more complex, leading to an even more difficult classification between normal and abnormal behavior.

Zhang and Lee [6] proposed the first (high-level) specific for ad hoc networks IDS approach. They proposed a distributed and cooperative anomaly-based IDS, which provides an efficient guide for the design of IDS 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. Huang and Lee [7] extended their previous work in a cluster-based IDS, in order to combat the resource constraints that MANET face. They use a set of statistical features that can be derived from routing tables and they apply the classification decision tree induction algorithm C 4.5 in order to detect normal vs. abnormal behavior.

Deng et. al. [8] 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 the Support Vector Machines (SVMs) classification algorithm. They use a set of parameters derived from the network layer and suggest that a hierarchically distributed approach may be a more promising solution versus a completely distributed intrusion detection approach.

Kachirski and Guha [9] proposed a cluster-based intrusion detection system built on a mobile agent framework. The proposed system uses mobile agents, with each agent performing a particular role, either monitoring, or decision or action. A few nodes, chosen by a distributed algorithm, 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 is minimized.

Liu et. al. [10] 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 [11] on feature vectors constructed from the training data.

Kannadiga et. al. [12] discussed the challenges and characteristics in pervasive computing devices and proposed an agent-based IDS to be deployed in pervasive computing environments. The proposed scheme is based on static and mobile agents and is designed considering the scarcity and heterogeneity of computing resources in pervasive environments.

B. *Watermarking in Information Security*

Watermarking is a mature research area that has been used extensively in information security. In particular, in the area of intrusion detection Wang et. al. [13] 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 only if the IDS subsystem has determined that there is an attack in progress.

Páez et. al. [14] 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 IDS based on mobile cooperative agents using watermarking techniques based on fingerprinting software.

Despite the important advantages that watermarking techniques 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 emergent Self-Organizing Maps 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 if we consider the fact that they have to combat the disadvantage of the resource constrained environment. Information visualization can help in the 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.

III. Intrusion Detection Model

A. *Intrusion Detection Architecture*

Malicious nodes in MANET may target to exploit features of the physical, MAC or network layers. The majority of the so far proposed security approaches in such networks has focused in the network layer, while little attention has been paid 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 [10], since it is placed 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

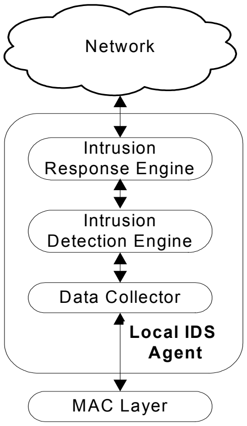

Fig. 1. Intrusion Detection Architecture

detection delay and the overall response time. Furthermore, MAC layer features make the discrimination between normal and abnormal behavior easier.

The architecture of the IDS applied to MANET can 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. Although cluster-based IDSs have the advantage of lower detection workload, the procedure of creating clusters and electing cluster heads may cause a large overhead. Moreover, the existence of cluster heads and the obvious possibility of their exploitation by malicious attackers might act as a single point of failure. 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, i.e. each node of the MANET should perform its local intrusion detection using local audit data. When the confirmation of other nodes to detect an attack is necessary, local intrusion detectors should cooperate.

The IDS architecture we adopt is composed of multiple local IDS agents as illustrated in Fig. 1, are responsible for detecting possible intrusions locally [6]. 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: is responsible for selecting local audit data and activity logs.

Intrusion Detection Engine: 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 the one 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 or abnormal.
- Perform watermarking in the local 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.

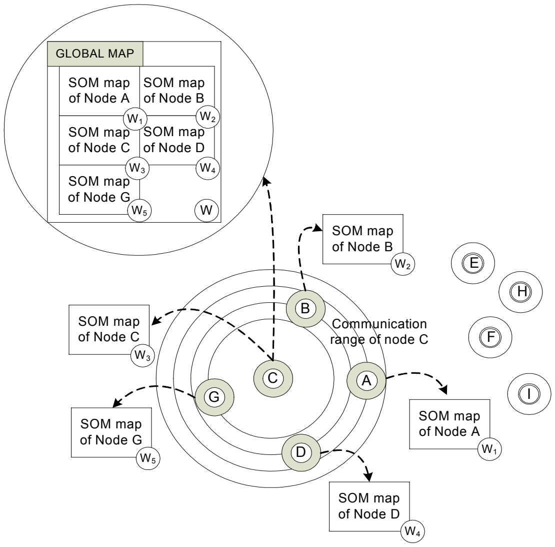

Fig. 2. Watermarked Emergent Self-Organized Maps of a MANET

In Fig. 2, nodes A, B, D and G are in the communication range of node C. Nodes A, B, C, D, G create their own eSOM U-Matrix (see section B below) and perform watermarking on them (illustrated as W_1, W_2, W_3, W_4, W_5 respectively). Node C selects the local watermarked eSOM u-Matrices from its neighbors and creates the global map of its local network. 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 it selects the appropriate node in order to forward its messages. In order to verify the authenticity and integrity of the global map node C performs also watermarking on the global map (illustrated as W in Fig. 2). Observing the local maps of all its neighboring nodes and by considering as secure the nodes that are not victims of attacks node C performs the selection of the appropriate node for the forwarding of messages.

Thus, each node selects the eSOM maps of its neighbors and uses them in order to have a view of the security of its neighbors, through the visual observation of the watermarked (not modified) maps produced by eSOM. After selecting the local maps from its neighbors each node creates the global map of its network consisted of all the local maps 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 Fig. 3.

Intrusion Response Engine: If the Intrusion Detection engine 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 MANET about the incident of intrusion. Local alarms are alarms sent to one hop neighbors of a node and global alarms are alarms sent to all nodes in a node transmission range. Moreover, in case that an intrusion is detected through the local eSOM map of a node, the attacked node is not selected to forward information to avoid possible loss of information.

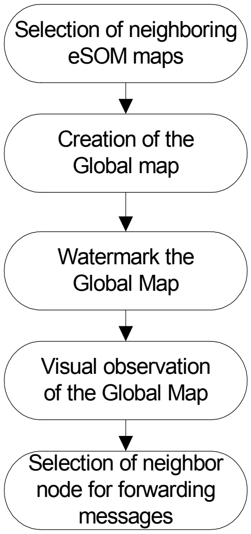
Selection of neighboring eSOM maps
Creation of the Global map
Watermark the Global Map
Visual observation of the Global Map
Selection of neighbor node for forwarding messages

Fig. 3. Procedure selection of forwarding node

Furthermore, in order to avoid possible flooding of notification messages, 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.

B. Intrusion Detection Engine based on Emergent Self Organizing Maps

Kohonen's Self-Organizing Maps (KSOM) [15] 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 Self-Organizing map 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.

Something that is often neglected in KSOM is that self-organization allows the emergence of structure in the data. According to [16], "Emergence is the ability of a system to produce a phenomenon on a new, higher level". In order to achieve emergence the existence and cooperation of a great number of elementary processes is necessary. Emergence may be present not only in natural but also in technical systems. One of the basic disadvantages of SOM maps is that their abilities are limited to a few neurons. On the other hand, emergent Self-Organizing Maps may expand to thousands neurons. A large number of neurons in eSOM are 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 emergent SOMs is performed by observing the whole emergent Self-Organizing Map and not by focusing on its neurons.

We have used the distance based (U-Matrix) method in order to *visualize* the structures generated by eSOMs. According to this method [17] 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 immediate neighbors on the map, *w(n)* is the weight vector associated with neuron *n*, then the height U-height of each neuron n is given by (1) [18]:

$$U - height(n) = \sum_{m \in NN(n)} d(w(n) - w(m)) \qquad (1)$$

where *d(x,y)* is the distance used in the SOM 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 small in areas that represent clusters, creating hills and valleys correspondingly.

In our MANET examples we trained emergent SOMs with logs of network traffic selected from a simulated MANET (using ns-2) and used eSOM U-matrices [19] in order to perform intrusion detection. In this 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 is 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 is created after the training of the emergent SOM, 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 the authenticity and detect any modifications of the maps. Watermarking is proposed as such a technique in this paper.

C. Protecting eSOM maps with Watermarking Techniques

Watermarking techniques have been mainly applied to protect the copyrights of digital media by embedding a unique message within the original information [3]. One of the most important requirements of watermarking is the perceptual transparency between the original work and the watermarked. 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 not watermarked image, in our case the eSOM U-Matrix.

We use watermarking techniques for the eSOM U-matrices, which are in the form of images in uncompressed format (bmp). We use the Lattice and the Block-Wise embedding methods [3]. The Lattice method has two parameters, alpha0 (lattice spacing) and beta (embedding strength), while the Block-Wise method has only one parameter alpha (quantization factor). We combined these two watermarking techniques in order to implement a cryptographic encoder-decoder that can be used in order to authenticate the nodes of a MANET.

For a fair comparison between the original and the watermarked work there are efficient distortion metrics [5]. 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, the so called *Watson perceptual distance* [4]. 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 I gives the metrics that are used more often.

The image we have used to perform our experiments is in bitmap grayscale format with 256x400 resolution. To observe the difference between the original and the watermarked image it is necessary to use the quality measures of Table I ([3], [4]) and calculate the ideal values. The ideal values of the test image (Fig.4) are presented in Table II.

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

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.

TABLE I
QUALITY MEASUREMENTS

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 Ratio (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 (NC)	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.

TABLE II.
IDEAL VALUES OF THE TEST IMAGE

MSE	0
SNR	94
PSNR	110
IF	100
NCC	1
CQ	138.178
WD	0

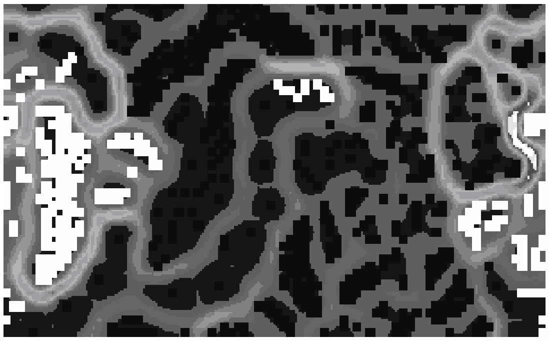

Fig. 4. Test Image -Emergent SOM U-Matrix of a node of a MANET

The reference marks are orthogonal to one another. 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 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 bits. 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 8x8 pixels in order to host the bits. The reference pattern consists of 8x8 random pixels. The pixel values are normalized to have zero mean and unit variance. Each bit is embedded by correlating a block against the 8x8 reference pattern, and by quantizing the result to an odd or even integer. The reference pattern, that is added to the 8x8 block according to the index of the closest point in the sublattice ($z_m[i]$), is computed by the following formulas:

$$l[i] = \frac{c_i * w_r}{|w_r|}, \qquad (2)$$

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

$$z_m[i] = 2\left\lfloor \frac{l[i]/(\beta|w_r|) - m_c[i]}{2} + 0.5 \right\rfloor + m_c[i], \qquad (3)$$

where m_c[i] is the corresponding message and the added pattern w_{a0i} is given by: $w_{a0i} = a_0(\beta z_m[i] w_r - c_i)$. (4)

The parameters in the embedding process are: a_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 (5) and then the least significant bit of it is detected. The coded message is then decoded with the trellis decoder:

$$z[i] = \left\lfloor \frac{c_i * w_r}{\beta w_r * w_r} + 0.5 \right\rfloor \qquad (5)$$

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 process respectively. The predefined parameters are a strength parameter alpha (a), which is used as the scaling factor of the luminance quantization matrix.

Four bits are embedded in the high-frequency DCT of each 8x8 (64 pixels) block in the image. In 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 in order to avoid any visual differences that would lead to unacceptably poor fidelity. Specifically 28 coefficients are used, which means that each bit is embedded in seven coefficients.

The seven coefficients that will host one bit are chosen randomly according to a seed number (see (6)). So 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.

$$C_I[i] = \left\lfloor \frac{C[i]}{aq[i]} + 0.5 \right\rfloor, \tag{6}$$

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 XORs 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 $b_e=b$. If $b_e \neq b$, 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 the DCT coefficients. Then the equation for the result is given by:

$$C_W[i] = aq[i]C_{wI}[i] \,. \tag{7}$$

At the decoder the procedure is exactly the same. From each 8x8 block the least significant bit b_e is extracted from each of the seven coefficients and it is compared to the embedded one b. If the two bits are different, the corresponding block is not authenticated but is marked as corrupted.

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 providing a very high quality of the watermarked image. 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 so 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 both the quality of the image and the ability to notice the corrupted blocks have the same importance, it is essential to combine the two embedding methods.

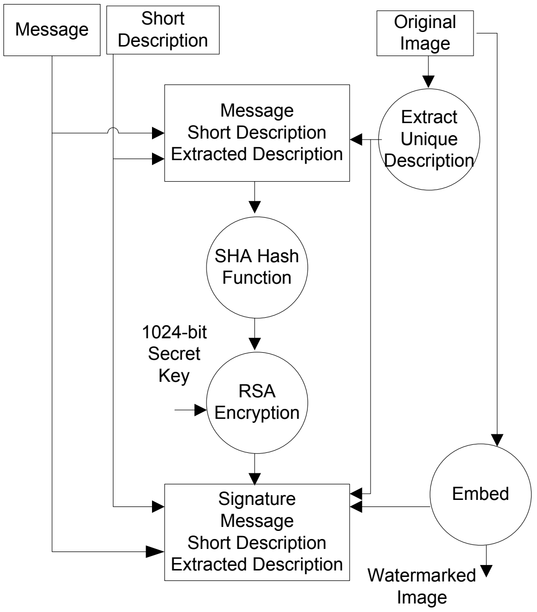
Message
Short Description
Original Image
Message Short Description Extracted Description
Extract Unique Description
SHA Hash Function
1024-bit Secret Key
RSA Encryption
Signature Message Short Description Extracted Description
Embed
Watermarked Image

Fig. 5. Cryptographic Encoder

The combination of the two embedding methods can be implemented in a cryptographic encoder-decoder. A node can give a short message like its ID and a short description (i.e. 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 entire three messages are inserted in a hash function and, then, the value is encrypted with an RSA 1024-bit secret key. The signature with the short and the extracted 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 Fig. 5.

From the watermarked version of the image, at the decoder's side, the signature, the short and the unique description are extracted with the Lattice method, while the message is extracted with the Block-Wise method. Then, a unique description is evaluated again and it is compared 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 low boundaries have been determined based on the ideal values of test image (table II). 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 with the one calculated at the decoder. If both the stages of the hash values and the unique descriptions are valid, the authentication process is

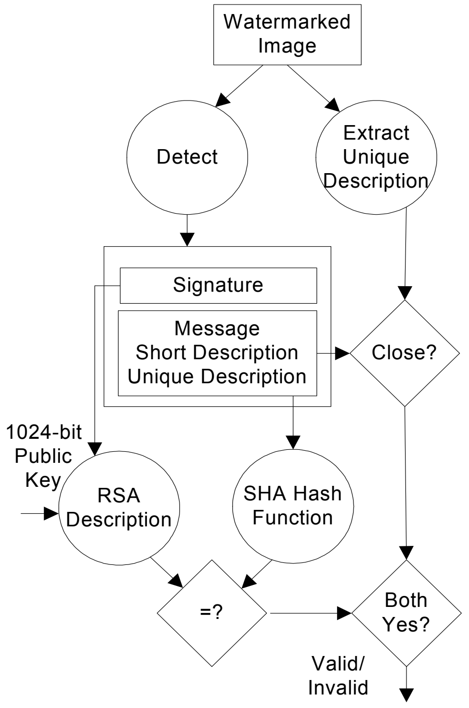

Fig. 6. Cryptographic Decoder

successful. The whole design of the decoder is presented in Fig. 6.

IV. Performance Evaluation

A. Intrusion Detection Engine Results

To evaluate the feasibility of the proposed intrusion detection engine we have conducted a series of experiments. For the experiments we have made the assumption that the network has no pre-existing infrastructure and that the employed ad hoc routing protocol is Ad-hoc On-demand Distance Vector (AODV).

We implemented the simulator using the ns-2 library. The simulation modeled a network of 50 hosts placed randomly within a 1800 x 1000 m^2 area. Each node has a radio propagation of 250 meters and the channel capacity was 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, then randomly selects and moves towards a destination with a speed uniformly distributed 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 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 stationary node.

We evaluated the performance of the 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 50 and 200 sec. The nodes perform normally between 0 and 50 sec. These parameters result in a network with rather high mobility and high traffic activity.

On average, twenty traffic generators were developed to simulate a Transmission Control Protocol (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 was 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 was randomly 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 the 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., there is 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 may be caused either by malicious behavior or by mobility. To add to the problem, we let the malicious node exhibit malicious behavior when it is most advantageous to it and not from the beginning of its participation in the network.

The statistical features we have used have been introduced by Liu et. al. [10] in their proposed approach for performing intrusion detection in the MAC layer. These features are:

- *Network allocation vector (NAV):* it is a node-specific characteristic, which depicts the time that the node will occupy the medium to send its messages.
- *Transmission traffic rate:* indicates the rate of the transmitted packets.
- *Reception traffic rate:* indicates the rate at which packets are received.
- *Retransmission rates of RTS packets:* indicates the rate of the ReadyToSend 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:* 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:* represents the number of neighbor nodes that have data transmission activities.
- *Forwarding node count:* represents the number of neighbor nodes that communicate directly with the monitoring node.

In order to avoid having a great influence of the attributes of some input vectors it is necessary to normalize the input data. We have normalized the data with mean zero and variance one. For the evaluation experiments we have used the Databionics eSOM tool ([18], [19]).

The presented evaluation shows that we can achieve a differentiation between normal and abnormal behaviors concerning packet-dropping attacks. In order to perform clustering with eSOM U-Matrices we followed the proceeding 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 Fig. 4. As it can be clearly seen the training data set has been divided in two classes that are very well distinguished, normal data class (dark color) and 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 any new mobility conditions.

To evaluate the efficiency of the proposed intrusion detection engine we use the Detection rate and the False alarm rate:

Detection rate $=\frac{TP}{TP+FN}$, False alarm rate$=\frac{FP}{TN+FP}$, where TP is the number of true positives, TN the number of true negatives, FP the number of false positives and FN the number of false negatives. The most effective approach should reduce as much as possible the False alarm rate and at the same time increase the Detection rate.

Fig. 7 presents the average Detection rate of all the source nodes that present traffic activity and are recognized as normal or attack by eSOM for a range of used pause times. The detection rate does not seem to be influenced by the mobility and in all cases it is over 80%. For long pause times the rate slightly decreases which is due to the TCP traffic and the degradation of the mobility. Indeed, a TCP agent stops sending data packets when it doesn't receive acknowledgments. Even after AODV discovers a new path to that destination, the agent keeps sending data packets through the malicious node, as the latter responds normally to control packets. As the network exhibits a rather low mobility, traffic is always rejected by the malicious node and it is soon stopped by the TCP agent, resulting in a degradation of the audit data fed to eSOM.

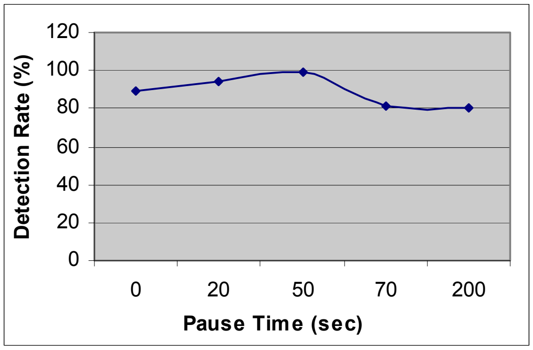

Fig. 7. Detection Rate vs. Pause Time

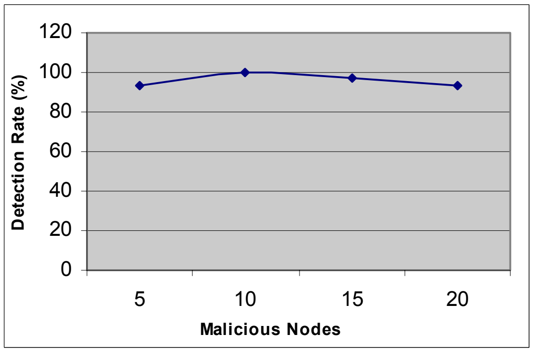

Fig. 8. Detection Rate vs. Number of Malicious Nodes

The detection rate as a function of the number of malicious nodes is presented in Fig. 8. The rate is rather high and, as in the previous figure, always over 80%. When a few malicious nodes exist in the network, the connections that are influenced by them are also a few, since source nodes move randomly in the network. This results in duplicated lines in the audit data set which is fed to eSOM, resulting in a decrease in the detection rate. When the number of malicious nodes is high compared to the number of source nodes, the TCP connections generated automatically by ns are a few, leading to multiple duplicate lines in the audit data that is fed to eSOM, which explains the decrease in the detection rate. The TCP traffic is used as a more realistic one. Other data traffic types (e.g. CBR) is under future investigation.

Tables III and IV present the average false alarm rate as a function of the paused times and the number of malicious nodes, respectively. When a source node generates traffic to different destinations and one of these connections is influenced by malicious nodes, then eSOM finds it difficult to distinguish among normal and abnormal traffic. If this is combined with multiple duplicate lines in the audit data due to mobility, the large number of malicious nodes produces rather high false alarm rates.

Two representative works in the area of anomaly detection is Deng et. al. [8] and Liu et. al. [10]. Deng et. al. [8] in their anomaly detection approach in MANET propose the use of SVM (Support Vector Machines) in a completely distributed architecture and achieve a false alarm rate range from 3.5±5.8% to 20.85±8.03% for the

TABLE III.
FALSE ALARMS VS. PAUSE TIME

Pause time (sec)	False Alarm (%)
0	21
20	20
50	22
70	20

TABLE IV.
FALSE ALARMS VS. NUMBER OF MOBILE NODES

Malicious Nodes	False Alarm (%)
5	26
10	22
15	17
20	21

Black hole attack and Frequent False Routing Requesting (FFRR) attack.

Moreover, Liu et. al. [10] in their approach for the packet dropping attack using cross feature analysis although the false alarm rate is low 0.29%, the detection rate is also rather low 72%. As the packet-dropping attack is a rather difficult attack to combat, in order to achieve low false alarm the detection rate is also decreased.

The proposed intrusion detection engine presents a comparatively high detection rate with the advantage of the visual representation of the 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.

B. Watermarking Results

In order to evaluate the performance and the efficiency of the embedding methods, several tests were performed. Several cases were considered, each with a different variable parameter (table V).

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. This means that the areas in the eSOM U-Matrix that are illustrated in Fig. 4 with a light color, representing the attack data class, i.e. the packet dropping data class will be watermarked with the Block-Wise method while the rest of the eSOM U-Matrix (the normal data class (dark color)) will be watermarked with 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 is 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 V some results of the combination are given in order to compare them to those of the two methods when they are used individually. The table justifies that the combination produces quality measurements between the two methods. In Table VI the maximum number of bits that can be hosted in the image using the two embedding methods and a combination of them are presented.

In order to verify that a possible modification of the eSOM U-Matrix can be detected by the decoder, we performed a test using the eSOM U-Matrix of Fig. 4. In the watermarked version, the light area representing the existence of an 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

TABLE V.
RESULTS OF THE COMBINED EMBEDDING METHOD

alpha0=0.93, beta=1.0, **alpha=0.1**	**Lattice** **alpha0, beta**	**Block-Wise** **alpha**	**Combined** **alpha0, beta,alpha**
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

alpha0=1.53, beta=0.8, **alpha=0.2**	**Lattice** **alpha0, beta**	**Block-Wise** **alpha**	**Combined** **alpha0, beta,alpha**
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

TABLE VI.
MAXIMUM NUMBER OF EMBEDDED BITS

	Lattice	**Block-Wise**	**Combined**
Max Embedded Bits	400	6406	>=4100

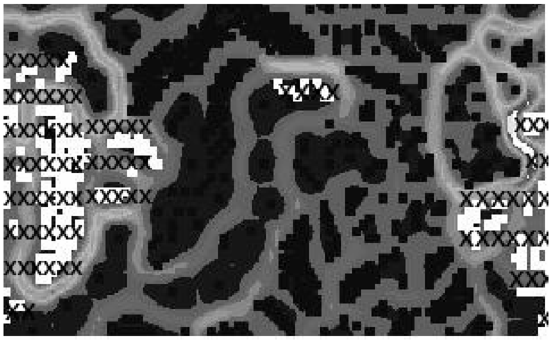

Fig. 9. Marked Image for the Test of the Cryptographic Encoder-Decoder

has failed. By observing Fig. 9 it is clear that the decoder has successfully located the modified blocks. Therefore, the implementation of the cryptographic encoder-decoder was correct.

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 watermarking technique are pre-calculated. Thus, the only computational complexity derives from the generation and verification of the digital signature. Furthermore, 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.

V. Conclusions And Future Work

We have presented an intrusion detection engine that is part of a local IDS agent existing in every node of a MANET. All the local IDS agents collaborate in order to compose an IDS for MANET. The proposed intrusion detection engine is based on emergent SOMs an efficient class of neural networks that produces as output a visual representation of the classification performed. We exploited the advantage of visualizing the network traffic and examined the classification performance of eSOM, for normal and abnormal behavior in MANET based on MAC layer features. Using eSOM, each node of the MANET creates its local eSOM map as well as the global map of its neighbors. Thus, each node has the option to select a secure routing path for packet forwarding by avoiding compromised neighbors. We should note here, that in order to combat the short duration membership problem of MANET, the proposed intrusion detection approach (employing the eSOM) should be trained in regular time periods. Thus, the trained dataset will depict the corresponding traffic conditions considering the connectivity and mobility issues in each time period of the MANET.

For the authentication of the local and the global maps an innovative and efficient watermarking method was proposed, which derives from the combination of two watermarking embedding methods, the Lattice and the Block-Wise. The proposed combined 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.

In this paper, we exploited the significant advantages of visual representation and watermarking, two research areas that have not previously used in MANET. We demonstrated the advantages of eSOM and visual representation in order to achieve intrusion detection. Furthermore, we used a novel watermarking technique for the authentication of the produced eSOM maps. For future work, the proposed intrusion detection engine can be employed to various routing protocols and used for the detection of various types of attacks as well as test it in real MANET applications and large-scale ad hoc networks.

Acknowledgment

This work has been partially supported by the Greek Research and Technology Secretariat under a PENED grant.

References

[1] S. Makki, N. Pissinou, H. Huang, "The Security issues in the ad-hoc on demand distance vector routing protocol (AODV)", In Proc. of the 2004 International Conference on Security and Management (SAM'04), pp.427-432.

[2] N. Komninos, D. Vergados and C. Douligeris, "Detecting Unauthorized and Compromised Nodes in Mobile Ad-Hoc Networks", Journal in Ad Hoc Networks, Elsevier Press, Vol. 5, (3), April 2007, pp. 289-298.

[3] J. Seitz, "Digital Watermarking for Digital Media", Information Science Publishing, Information Resources Press Arlington,VA, USA, 2005.

[4] Q. Zhang, "New techniques for Digital Watermarking", ProQuest / UMI, 2006-12-13.

[5] T. Furon, "A survey of Watermarking Security, Digital Watermarking", In Proc. of 4h International Workshop on Digital Watermarking (IWDW'05), Sept. 2005, Siena, Italy, pp. 201-215.

[6] Y. Zhang, W. Lee, Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks", Wireless Networks vol. 9, no. 5, pp. 545-556, Sept. 2003.

[7] Y. Huang, and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks", In Proc. of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), October 2003, Fairfax, Virginia, USA, pp. 135-147.

[8] H. Deng, Q. Zeng, and D. P. Agrawal, "SVM-based Intrusion Detection System for Wireless Ad Hoc Networks", In Proceedings of the IEEE Vehicular Technology Conference (VTC'03), Oct. 2003, Orlando, Florida, USA, pp. 2147-2151.

[9] O. Kachirski, and R. Guha, "Intrusion Detection Using Mobile agents in wireless Ad hoc Networks", In Proc. of the IEEE workshop on Knowledge Media Networking, July 2002, Kyoto Japan, pp.153-158.

[10] Y. Liu, Y. Li, H. Man, "MAC Layer Anomaly Detection in Ad Hoc Networks", In Proceedings of 6th IEEE Information Assurance Workshop, June 2005, New York, USA, pp.402- 409.

[11] Y. Huang, W. Fan, W. Lee, P.Yu, "Cross-Feature analysis for Detecting Ad-Hoc Routing Anomalies", In Proc. of the 23rd International Conference on Distributed Computing Systems, May 2003, Rhode Island, USA, pp. 478.

[12] P. Kannadiga, M. Zulkernine, and S. Ahamed, "Towards an Intrusion Detection System for Pervasive Computing Environments", In Proc. of the International Conference on Information Technology (ITCC'05), Las Vegas, Nevada, April 200, pp. 277-282.

[13] X. Wang, D.S. Reeves, S.F. Wu, J. Yuill, "Sleepy watermark tracing: an active network-based intrusion response framework", In Proc. of the IFIP 16th International Conference of Information Security, June 2001, Paris, France, pp. 369-384.

[14] R. Páez, C. Satizábal, J. Forné, "Cooperative Itinerant Agents (CIA): Security Scheme for Intrusion Detection Systems", In Proc. of the International Conference on Internet Surveillance & Protection (ICISP '06), August 2006, Côte d'Azur, France, pp. 26.

[15] S. Haykin, "Neural Networks: A comprehensive Foundation", Prentice-Hall, New Jersey, USA, 2nd edition, 1999.

[16] A. Ultsch, "Data Mining and Knowledge Discovery with Emergent SOFMs for Multivariate Time Series", In Kohonen Maps, Elsevier Science, pp. 33-46, 1999.

[17] A. Ultsch, "U*-Matrix: a Tool to visualize Clusters in high dimensional Data", University of Marburg, Department of Computer Science, Technical Report, Nr. 36, December 2003.

[18] A. Ultsch, F. Moerchen "ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM", Tech. Report Dept. of Mathematics and Computer Science, University of Marburg, Germany, no.46, 2005.

[19] Databionic ESOM Tools, Available from <http://databionic-esom.sourceforge.net/devel.html>