Visual-Assisted Wormhole Attack Detection for Wireless Sensor Networks

Eirini Karapistoli[1(✉)], Panagiotis Sarigiannidis[2], and Anastasios A. Economides[1]

[1] Interdepartmental Programme of Postgraduate Studies in Information Systems, University of Macedonia, Egnatia 156, Thessaloniki, Greece
{ikarapis,economid}@uom.gr
[2] Department of Informatics and Telecommunications Engineering, University of Western Macedonia, Karamanli and Ligeris Street, 50100 Kozani, Greece
psarigiannidis@uowm.gr

Abstract. Wireless sensor networks (WSNs) are gaining more and more interest in the research community due to their unique characteristics. In addition to energy consumption considerations, security has emerged as an equally important aspect in their network design. This is because WSNs are vulnerable to various types of attacks and to node compromises that threaten the security, integrity, and availability of data that resides in these networked systems. This paper develops a powerful, anomaly detection system that relies on visual analytics to monitor and promptly detect a particularly devastating form of attack, the *wormhole attack*. Wormhole attacks can severely deteriorate the network performance and compromise the security by disrupting the routing protocols. The proposed system, called VA-WAD, efficiently utilizes the routing dynamics to expose an adversary conducting a wormhole attack. Then, the output of the anomaly detection engine feeds the radial visualization engine of VA-WAD, which further assists the understanding and analysis of the network topology improving the detection accuracy. By employing an outer ring, VA-WAD also records the network security events occurring in the WSN on a 24 h basis. The obtained simulation results demonstrate the system's visual and anomaly detection efficacy in exposing concurrent wormhole attacks.

Keywords: Wireless sensor networks · Wormhole attacks · Anomaly detection · Security visualization

1 Introduction

A wireless sensor network is a network of cheap and simple processing devices (called sensor nodes) that are spatially distributed in an area of interest in order to cooperatively monitor physical or environmental conditions and transmit the collected information to a remote server for further processing [1]. Most of the

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 222–238, 2015.
DOI: 10.1007/978-3-319-23829-6_17

applications in Wireless Sensor Networks (WSNs) are envisaged to support the remote and unattended operation of a large number of sensor nodes. In such a setting, efforts to extend the network lifetime are of crucial importance. Besides energy consumption considerations, security is an equally critical component that contributes to the performance of WSNs. The major challenges that need to be dealt with in addressing security issues mainly stem from the open nature of the wireless medium and the multi-hop cooperative communication environment. These factors make network services more vulnerable, specifically due to attacks originating from within the network [2–4].

Routing protocols in WSNs are susceptible to numerous attacks. A detailed survey of security attacks can be found in [5]. In this paper, we focus on a particularly devastating form of attack, the *wormhole attack* [6,7]. A wormhole attack is a special type of collusion attack on sensor networks in which two colluding malicious nodes use wormhole links to capture and replay communicated messages in order to disrupt the network protocol. To launch a wormhole attack, the colluded malicious nodes establish a direct communication channel between themselves bypassing several intermediate nodes. The established channel can be an out-of-band high-speed communication link or an in-band logical tunnel. Once established, the wormhole link attracts most of the traffic since the control packets traversing through a wormhole link advertise a much better link metric. Selection of such links results in denial-of-service (DoS), affecting the performance of the network severely. It is even possible to occur more than once wormhole links making the problematic situation yet harder. It has been shown that a strategic placement of the wormhole can disrupt on average 32 % of all communication across the network [8].

A number of security solutions have been proposed to deal with these attacks [7,9,10]. Most of these defensive methods, however, require the sensor nodes to be equipped with some special hardware, such as location-finding devices (Global Positioning System, GPS), synchronized clocks, or directional antennas. Hence, such methods are limited in their efficacy owing to high computational resource requirements and communication overhead. Recently published wormhole detection algorithms [11–13] overcome this problem by relying solely on neighborhood or connectivity information. Nevertheless, these automated tools are limited in their efficiency owing to computational resource requirements and incurred communication overhead, but most importantly, they lack the *reasoning ability* that is crucial for making decisions about anomalous data that may or may not be a threat, with the typical consequence of a high false positive rate.

Since wormhole attacks are dynamic, if analysts cannot absorb or properly correlate the network traffic data, it will be difficult for them to detect them. Developing tools that increase the situational awareness of all those actions responsible for the network's safe operation can increase the network's overall security. System administrators are typically limited to textual or simple graphical representations of network activity. Information visualization instead, has effectively increased operators' situational awareness, letting the security professionals to more effectively detect, diagnose, and treat anomalous conditions.

A growing body of research validates the use of visualization to solve complex data problems [14,15]. Visualization elevates information comprehension by fostering rapid correlation and perceived associations. To that end, the display's design must support the decision-making process by identifying problems, characterizing them, and determining appropriate responses [16].

Our visualization technique integrates information from network log files into an intuitive, flexible, extensible, and scalable visualization tool, called VA-WAD, that presents critical information concerning network activity in an integrated manner, increasing the user's situational awareness. VA-WAD tackles the wormhole detection problem in large-scale WSNs by relying on the dynamics of concentric circles. To help address the security visualization challenges, VA-WAD offers the following contributions;

- A novel wormhole attack detection engine that relies on topological comparisons to timely detect and resolve multiple instances of wormhole attacks that are present in the WSN.
- A powerful visualization engine that uses a novel, cross-free radial layout to monitor the evolving status of the network and to efficiently reveal active wormhole links. It consists of the *planar view*, which uses concentric circles that expand outwards radially to visualize the network topology, and the *event logger* that keeps track of the network events on a time-adjusting basis.

The remainder of the paper is organized as follows. Section 2 reviews existing security approaches aimed at detecting wormhole attacks launched against WSNs. Section 3 introduces the anomaly detection and visualization engines of VA-WAD. In Sect. 4, the detection accuracy and visual efficacy of the VA-WAD system are evaluated through a simulated attack scenario. Finally, Sect. 5 concludes the paper and discusses future extensions.

2 Related Work

There are several potential ways of defending against wormhole attacks, each of which exploits a different unique feature exhibited by a wormhole link. Generally speaking, these methods can be categorized in two broad categories;

Automated Approaches: Most of the existing schemes [10] exploit the abnormal length of a wormhole. As previously stated, a wormhole link is usually established between nodes that are physically separated by a large distance, thereby bypass several intermediate nodes. Therefore, the simplest way to defend against a wormhole attack is by preventing nodes from being tricked into forming a wormhole link through equipping nodes with GPS and verifying the relative position of a transmitter during peer (link) establishment. Location-based schemes can successfully defend external wormhole attacks, but cannot prevent Byzantine wormholes [17,18] from being established as the colluded nodes involved in the attack are legitimate part of the network.

The other unique characteristic of a wormhole is that it abnormally increases the node's neighborhood, and this feature is being exploited in [12] to detect hidden wormholes. Let W_1 be a wormhole node that shares an out-of-band channel with another wormhole node W_2. Now, W_1 can relay on its neighborhood information to W_2 and trick W_2's neighbors into believing that they share direct neighborhood with W_1's neighbors. This abnormally increases the neighbor count of a node-sharing neighborhood with a wormhole node. Unfortunately, such schemes fail to detect Byzantine wormholes as the link being established between colluded internal nodes does not alter the neighborhood information of their respective neighbors. On similar lines, protocols exist that exploit abnormal path attractions of wormhole nodes [11].

Visual-based Approaches: On the visualization frontier, two schemes have been proposed to address the problem of visual-based wormhole detection in WSNs. Early in 2004, Wang and Bhargava [19] proposed a security enhancing visualization mechanism for WSNs, called MDS-VOW, which is capable of identifying the occurrence of a wormhole attack in stationary wireless sensor networks. Using multi-dimensional scaling (MDS) and a surface smoothing strategy, a virtual layout of the network is computed. The shape of the reconstructed network is then analyzed. If any wormhole exists, the shape of the network will bend and curve towards the wormhole, otherwise the network will appear flat. Later on, Wang and Lu [20] extended the MDS-VOW concept proposing an improved detection mechanism, called interactive visualization of wormholes (IVoW). IVoW efficiently integrates automatic intrusion detection algorithms with visual representation and user interaction to support visualization of several wormholes in large-scale dynamic WSNs. While promising, the approaches of this category require greater visualization effort in order to come up with a firm final resolution as well as a more insightful human-computer interaction [21].

As apparent, the existing security solutions (both automated and visual-based) are either limited in their efficiency owing to computational resource requirements and communication overheads or can only deal with a single wormhole attack instance that is present in the network. Compared to the previous security mechanisms, VA-WAD encompasses the strengths of both automated and visual-based approaches in order to accurately and promptly detect concurrent wormhole attack instances in WSNs. In the subsequent sections, we describe in detail the anomaly detection and visualization engines of VA-WAD.

3 VA-WAD: A Visual-Assisted Wormhole Attack Detection System for WSNs

VA-WAD is a system that fully leverages the power of both visualization and anomaly detection analytics to guide the user to quickly and accurately detect wormhole attacks in large-scale WSNs. The VA-WAD system builds on two core components; the *wormhole anomaly detection engine* (WAD), and the *visualization engine*. The WAD component represents the system's automated anomaly

detection logic, while the visualization engine, is the projection tool. We begin our analysis by stating the network assumptions, and then we describe in details the two main components of VA-WAD.

3.1 Model Assumptions

In the present work, we consider a typical WSN comprised of a large number of autonomous sensors that are spatially distributed in an area of interest in order to support a security-oriented application. A snapshot of the simulated topology is shown in Fig. 1. In such a setting, the sensor nodes remain more or less static for the duration of the deployment. Moreover, the legitimate sensors establish secure peer links [22] and forward the sensed data to the Base Station (BS), which is a typical reporting method in WSNs. The communication between the nodes is based on a flat routing scheme where all nodes are assigned equal roles, i.e., they are peer. The BS is responsible for collecting the control packets that are being traversed through the network. These packets contain various information such as the routing cost, the neighbor list, and the next hop of each node. We also assume that the WSN is protected during the initial deployment and setup phase due to the following reasons:

- In many cases, the initial deployment and routing setup takes place under supervision discouraging any malicious actions.
- The initial routing setup phase is completed when all nodes have determined the shortest path to the BS. In most cases, the duration of such a process could be performed within a short period of time impeding malicious actors to timely employ their attack.
- In cases of an unsupervised deployment, either the deployment location is temporarily unknown, e.g., in military applications, or the time needed for launching a wormhole attack is much longer than the initial routing setup.

Regarding the attacker model, we assume that during the simulation, randomly selected intelligent adversaries include themselves in the network by replicating (compromising) legitimate sensor nodes. An adversary is capable of estab-

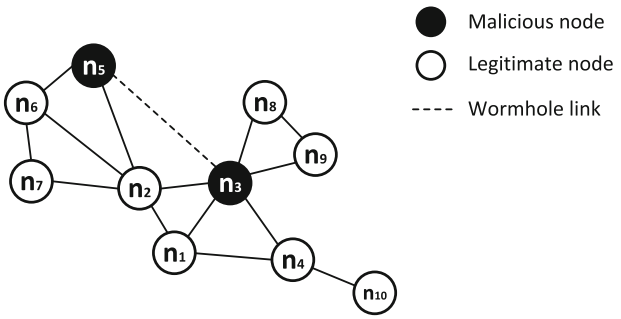

Fig. 1. The topology of a WSN with 10 nodes; nodes n_3 and n_5 form a wormhole link.

lishing a high-speed low-latency communication link, required to launch a hidden wormhole attack.

3.2 The Anomaly Detection Engine

The first major component of VA-WAD is the anomaly detection engine. The employed Wormhole Attack Detection (WAD) mechanism of this engine is tasked with capturing the routing dynamics of the network. Owing to the visualization need [21], the WAD mechanism is supposed to be centralized (meaning that the processing and decision making is done centrally) and is independent of the routing protocol in use. Since a wormhole attack intends to disorientate the routing protocol of the WSN, there is little doubt that it will leave signs of intrusion that affect several key routing features. In particular, the operation of a wormhole attack influences; (a) the neighbor list maintained by each node (n_l), (b) the next hop of each node (n_h), and (c) the routing cost to the sink node (r_c). Hence, the proposed detection mechanism monitors the aforementioned routing features to capture any suspicious actions. To accomplish this feat, WAD runs in phases; the *anchor phase*, the *monitoring phase*, the *detection phase*, and the *resolution phase*.

Anchor Phase: In this initial routing setup phase, which comes right after the WSN deployment phase, the WAD mechanism stores centrally a set of values for each sensor node relative to the routing process. Having in mind that any applied routing protocol provides the nodes with the optimal routing cost to the BS r_c^{opt} along with its next hop node n_h^{opt}, the mechanism takes the opportunity to record these "optimal" routing values of each node, also known as *anchor values*. The mechanism makes use of this information to construct the *list of the intermediate routing nodes*, i_{nl}. This list contains the route a packet follows to reach the BS based on the next hop neighbor of each node. These three features are utilized by WAD to facilitate the detection of wormhole attacks.

Monitoring Phase: The second phase declares the monitoring period. Under normal operation, the nodes uninterruptedly and periodically exchange routing information in a neighbor-to-neighbor basis following the routing updates of the adopted protocol. Normal operation of the WSN implies that all nodes advertise either routing cost equal to the anchor cost or a larger one, in case for example an intermediate routing node stops functioning due to battery exhaustion. If a node advertises an updated routing cost r_c^{new} less than its anchor cost r_c^{opt}, then the mechanism pinpoints the suspicious change and switches to the detection phase.

Detection Phase: The third phase aims at detecting the adversary nodes forming the wormhole link. The detection algorithm acts as follows; first, it creates an *affected node list*, a_{nl}, by inserting those nodes that advertised reduced

routing costs in it. Apparently, one of the two nodes forming the wormhole link is included in this list. For the wormhole link to become attractive to the highest percentage of network traffic flows, one of the two ends of the link needs to deliberately choose its location such that it is closer to the BS. Keeping this fact in mind, the nodes forming the wormhole link exhibit the following properties:

- the malicious node that is part of the wormhole link, and is closer to the BS has a routing cost that is unchanged before and after the attack.
- the malicious node that is part of the wormhole link and is farther from the location of the BS, advertises smaller routing cost after the attack, and as such it is included in the *affected node list*, hereafter, designated as *source node*, n_s.

The detection phase targets the filtering of the *affected node list* in order to expose the source node. A new list is thus produced, called *critical node list*, c_{nl}, in accordance to the following criterion: "since nodes that belong to the affected node list present reduced routing cost as an outcome of the attack, these nodes subtly include the source node within their list of intermediate routing nodes". Hence, the source node is present in the list of every affected legitimate node. To this end, the detection algorithm examines those nodes that exist in every list of intermediate routing nodes, i_{nl}, and creates the *critical node list*, c_{nl}. Lastly, the algorithm selects the node having the minimum routing cost, r_c^{min} within the *critical node list* to be the source node, n_s. Apparently, the node in the other end of the wormhole link is the next hop node of the *source node*.

Algorithm 1. The Wormhole Attack Detection (WAD) Mechanism

{ Anchor Phase }
Initialize the following lists; i_{nl}, a_{nl}, and c_{nl}
for each routing setup information coming from node i in the network **do**
  update the i_{nl} list with the following data r_c^{opt}, n_h^{opt} associated with node i
end for
{ Monitoring Phase }
for each routing update coming from node i in the network **do**
  update the routing cost, r_c^{new} of node i
end for
{ Detection Phase }
if $r_c^{new} < r_c^{opt}$ **then**
  insert node in *affected node list*, a_{nl}
end if
for every possible pair of nodes in the *affected node list*, a_{nl} **do**
  compare the i_{nl} and a_{nl} lists and create the *critical node list*, c_{nl}
  select the node having the r_c^{min} within the c_{nl} list to be the n_s
end for
{ Resolution Phase }
black list and isolate the source node, n_s and its next hop.

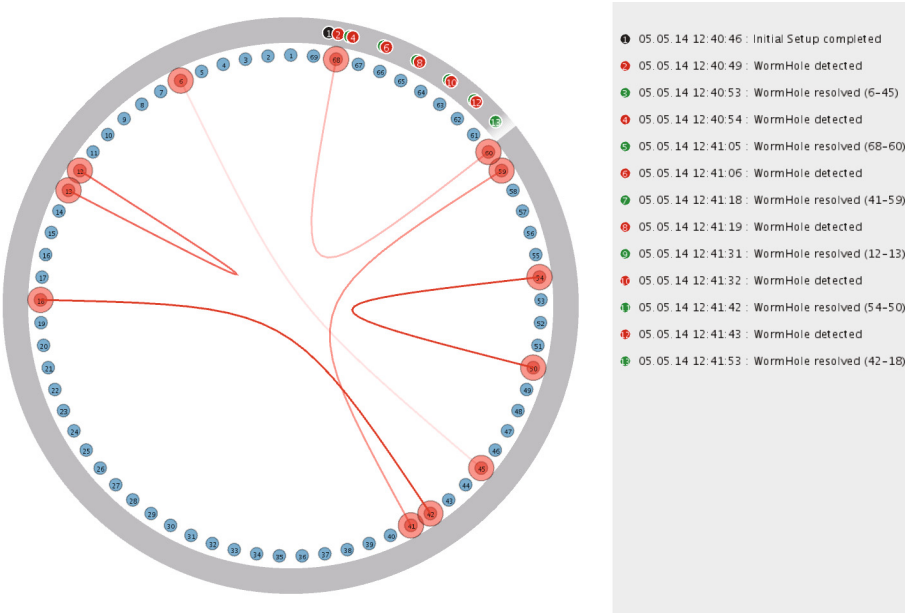

Fig. 2. The visual interface of VA-WAD consisting of the *Planar View* (on the left), and the *Event Logger* (on the right).

Resolution Phase: During the final phase, the WAD mechanism attempts to mitigate the wormhole attack. Upon detecting the pair of malicious nodes constituting the two ends of the wormhole link, the mechanism black lists these nodes, and re-initializes the routing process. Following this, the pair of malicious nodes is isolated from the routing process. It is worth mentioning that the anchor values are kept untouched during the routing reset process. After that, the mechanism returns to the monitoring phase, and keeps monitoring the WSN to detect other potential wormhole threats.

3.3 The Visualization Engine

The WAD engine is complemented with the visualization engine enriching the VA-WAD system with simple, but powerful visual forms in order to provide the user with real-time, informative, and accurate views of the evolving network status in an animated fashion. Figure 2 illustrates the main Graphical User Interface (GUI) of the visualization engine of VA-WAD. As it can be seen, the entire screen space of VA-WAD is divided into two sections; the Planar View (on the left), and the Event Logger (on the right). The width ratio of the two regions is defined as 7:3. Next we describe, each of these components in detail.

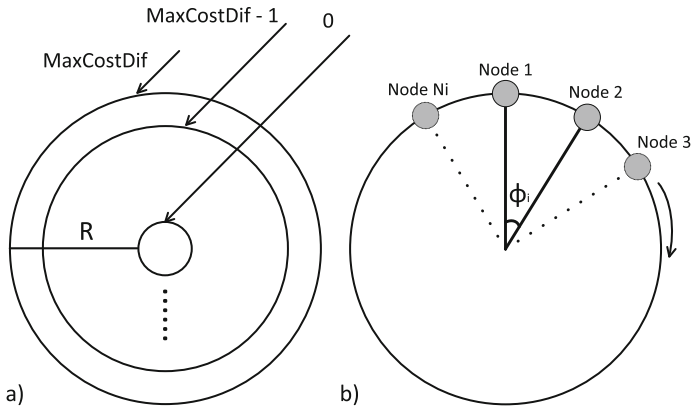

Fig. 3. (a) The structure of the Core Circle, (b) the orbital placement of the nodes on each ring defined by the parameter φ_i.

The Planar View: The *Planar View* is further divided into the *Core Circle*, and the *Time Ring*. This outer ring is used in order to enhance the visual dynamics with time domain information. On the other hand, the Core Circle constitutes the main visualization engine that interprets and projects the outcome of the WAD mechanism utilizing effective visual analytics techniques. Next, the above views are introduced and described in detail.

The Core Circle: The *Core Circle* defines the fundamental visualization mechanism. In essence, it constitutes a fully dynamic visualization environment that endeavors to expose and resolve wormhole attacks on a real time basis by continuously reconstructing the network topology that is laid out on a radial layout [23]. The visualization places the circles in insightful orbital positions and dynamically changes the morphology of the inner rings. Inspired by the comparative analysis between Cartesian and radial variants of information visualization performed by Diehl *et al.* [24], VA-WAD utilizes a radial graph layout for arranging thousands of sensor nodes in the screen space. Each legitimate node is represented by a circle colored in light blue [25], having the node ID drawn in the center of the circle with a black color.

Figure 3 shows the structure of the *Core Circle*. The circle radius is denoted by R. Similar to the detection and resolution mechanism, the visualization interface obtains routing dynamics and generates the inner rings. The nodes, shaped in circles, are suitably placed on the ring perimeter. In this way, each circle defines an orbit on the ring. The maximum value of the routing cost differentiation amongst all nodes defines the number of inner rings, and it is represented by the *MaxCostDif* parameter. The routing cost differentiation expresses the difference between the current routing value, and the anchor routing value. The maximum value of this difference is calculated in order to demonstrate the magnitude of the tunnel length of a potential wormhole attack. This value is provided to the visualization platform by the WAD strategy. The system creates

$MaxCostDif + 1$ inner rings to place the nodes. The radius of the i-th ring is equal to:

$$r_i = \frac{R \times i}{MaxCostDif} \tag{1}$$

where i stands for the index of each inner ring. Each ring implies a specific value of routing cost differentiation. Nodes whose value is zero are placed on the most inner ring perimeter. On the other hand, the most outer ring carries those nodes that present the largest noticed routing cost differentiation. However, it is important to point that during the anchor phase, the strategy provides the environment with this value, which is continuously updated since the nodes improve their routing information trying to find the optimal value. At this phase, the inner rings simply construct a visual pattern. At the end of this phase, a distinctive visual pattern is obtained as an indication of the unique identity of the WSN under investigation. Upon completion of the anchor phase, the normal operation of the WSN is demonstrated by a single outer ring declaring that all nodes advertise a zero difference between the updated routing cost and the anchor cost value.

The placement of the circles on the rings' perimeter takes place in a simple, yet effective way avoiding occlusion effects. As depicted in Fig. 3, the number of nodes associated with each inner ring, i (denoted by N_i), is calculated for each time instant, and each node is placed in a unique position on the ring perimeter. In particular, the ring is divided into N_i sectors. The angle that defines each sector is determined by the parameter φ_i (in rads), which is given by the following expression:

$$\varphi_i = \frac{2\pi}{N_i} \tag{2}$$

The visualization engine follows the outcome of the WAD mechanism by producing informative network reconstructions and visual patterns that aid the security analysts to timely detect wormhole attacks. As it can be seen from the Fig. 2, the *Core Circle* provides the user with multiple levels of data details. To this end, the proposed visualization system applies the following visual forms:

- Upon the production of the *critical node list*, the system highlights the critical nodes, represented as circles with an outer red ring. Hence, the user can distinguish the candidate malicious nodes during the detection phase of the source node.
- Upon detection of the source node(s), the system uses a larger circle to represent the adversary. It also uses red colored link metaphors to highlight the two ends of the wormhole link. We used the bezier curves for this purpose. By doing this, the wormhole links remain highlighted until a user maintenance is performed and the attack is mitigated. Figure 2 shows six wormhole links (6–45, 68–60, 41–59, 12–13, 54–50, and 42–18) that are highlighted in this way. Note that each link has a different color variation. Supposing a number

of q links, the color of each link is a red one, having an alpha parameter, $A_i, 1 \leq i \leq q$, which is determined as follows:

$$A_i = \frac{255 \times i}{q}, \text{where } 0 \leq A_i \leq 255 \tag{3}$$

The Time Ring: The Core Circle is accompanied by a *Time Ring*, which enhances the visualization framework with crucial temporal information. The visualization projection produces messages that are placed on the time ring based on their generation time. In particular, the time ring shows an integrated view of the network activity, e.g., what happened in the past hour or past 24-hours or on any time-adjusting basis. A *time runner* is designed to pinpoint the time elapsed since the beginning of the operation of the visualization system. The runner is moving in a clock-wise manner while the zero-point (starting point) is considered the 12 o'clock point of the circle. The runner is moving in a different speed based on the time basis defined on the time ring. In a nutshell, the Time Ring is considered as an essential component of the visualization engine in view of the time domain provisioning.

The Event Logger: The *Event Logger* section is located at the right region of the screen. Its light grey background is used to divide the screen into two distinctive regions. Each message produced consists of a colored bullet, the generation date and the message text. The colored bullet that is placed before the text is colored in accordance to the message type. Three message types are defined, namely (a) the initial setup completion message, colored in black, (b) the wormhole detection message, colored in red, and (c) the message announcing the resolution of the attack in green. Furthermore, a serial number is kept for each message in order to facilitate the information interpretation on the user side. The event logger is automatically updated as the number of messages increases. This means that old messages are removed, and are replaced by new messages as soon as the screen space is full. As shown in Fig. 2, upon generation of a message, a relevant circle is placed on the time ring according to the generation time using the bullet's color. Thus, the generated events are directly linked with the visualization system through the colored circles on the time ring. Moreover, the event logger informs the user about the pair of detected malicious nodes. As it can be seen, Fig. 2 pinpoints six pairs of malicious nodes (6–45, 68–60, 41–59, 12–13, 54–50, and 42–18) and provides information about their detection and resolution time.

4 Performance Evaluation

In this section, we evaluate the visual and wormhole attack detection efficacy of the VA-WAD system. We used the OMNeT++ [26] environment in order to generate our simulation scenario (network topology and traffic), and to feed the VA-WAD system. We simulated a 802.15.4 peer-to-peer sensor network configured with the WAD detection scheme. A number of legitimate sensor nodes

(varying from 50 up to 100) were uniformly placed in the sensor area without inducing unconnected nodes. The nodes are considered to be stationary during the simulation. Each node has a communication range of radius $R = 50$ m. A number of wormhole links (up to six) were also introduced and were capable of launching an attack against the WSN. The routing update period was set to 1.5 s. In all investigated scenarios, it is considered that neither the malicious node nor the legitimate nodes are aware of the actual position of each other.

4.1 Visual Efficacy

Firstly, we validate the efficacy of the VA-WAD's visualization engine. We used the simulated attack scenario described above to generate the following figures. The sequence of figures that follow shows how a set of wormhole attacks emerges out of the background noise of the visual interface, assisting users to rapidly detect and identify wormhole attacks. Please note that for reasons of higher resolution the event logger is omitted in each of those figures.

Initially, the WSN is deployed. The nodes begin to identify their neighbors, and then, they apply the predefined routing protocol to identify their path to the BS. The anchor phase of the proposed WAD mechanism is thus active. After a short period of time, the initial routing setup process is finalized. For simplicity reasons, we consider a routing protocol that utilizes a hop count routing metric as a routing strategy. At this point, the nodes have found the shortest path to the sink node. The Planar View, as illustrated in Fig. 4a, shows the final reconstructed topology of the anchor phase. At this point, the visual pattern of the initial routing setup has been produced. For example, node 6 has been placed in the fifth inner ring. This means that node 6 experienced six-hops difference since the beginning of the network operation. Hence, it advertises to their neighbors that it has a routing cost equal to eight towards the sink node. The WAD mechanism records the anchor values.

The subsequent image shown in Fig. 4b dominates the planar view. It is what we get since the first event, i.e., the initial setup, has been finalized. The first event has been recorded in the time ring as well with a circle containing the number 1 (the event logger is not shown here due to space limitations). The visualization interface implies that the WSN is operating normally. The WAD mechanism then switches to the monitoring phase.

To demonstrate VA-WAD's ability to detect multiple wormhole attacks, in Figs. 5 and 6 we launch two concurrent wormhole attacks. The two wormhole links are created between nodes 6 and 45, and nodes 41 and 59. Figure 5a, illustrates the impact of these attacks which are captured by the dynamics of the concentric circles. Suppose that the WAD mechanism is in the detection phase. A new message has been generated indicating the presence of the adversary. At this point, the system is unaware of the number of attacks in the network. Actually, it knows that a wormhole attack is active, and as such, it tries to expose the source node(s). The visualization interface has marked with red a list of suspicious nodes 6, 7, 8, 18, 40, and 41. At least one of them is the source node of a wormhole link. Note that in Fig. 5a, node 6 is now located on the third ring

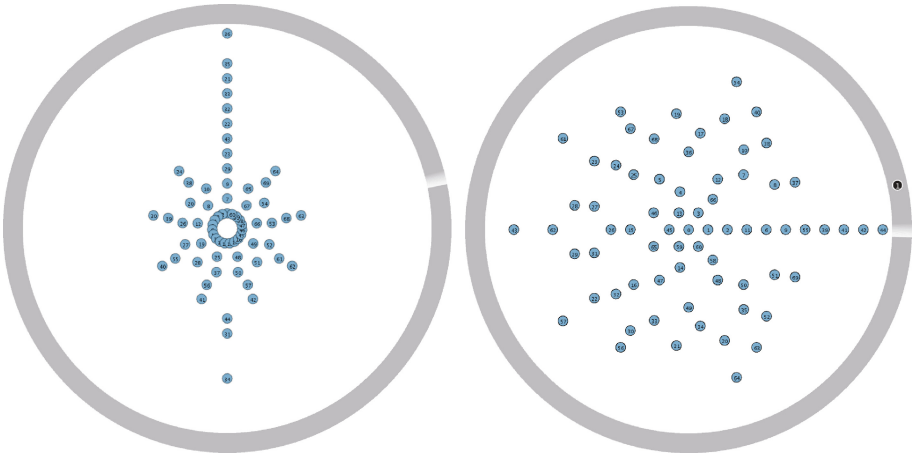

Fig. 4. (a) (Left) Network topology reconstruction prior to finalization of the initial routing setup. (b) (Right) Network in normal activity.

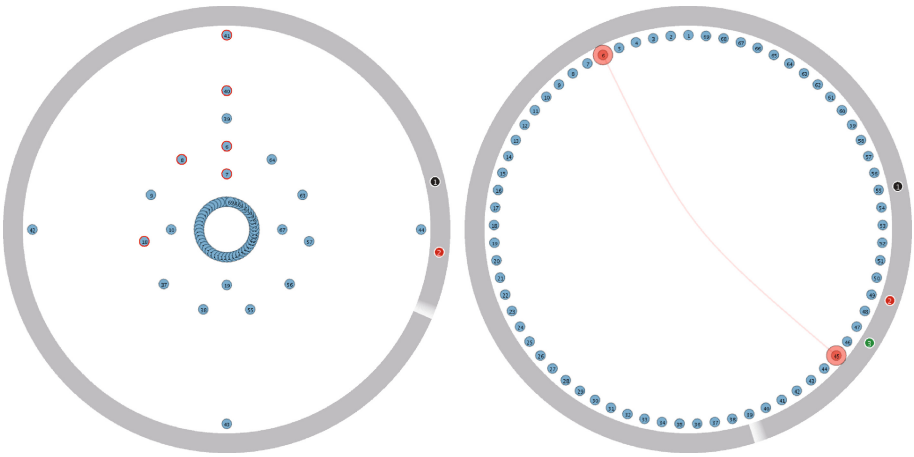

Fig. 5. (a) (Left) VA-WAD detected an abnormal activity. Critical nodes are marked in red color. (b) (Right) VA-WAD resolved a wormhole link between nodes 6 and 45 (Color figure online).

perimeter out of the total six rings. Following the above remarks, a user could interpret this visual information and conceive that node 6 now has a routing cost equal to 6-3 = 2, due to the presence of the wormhole link. A snapshot of the subsequent frame of the animation is illustrated in Fig. 5b. Indeed, nodes 6 and 45, which form the wormhole link, appear to be connected with the help of a link metaphor. We used a bezier curve to highlight the connection of the two ends of the wormhole. Following the identification and isolation of the wormhole link, the system returns to the monitoring phase. The routing information is updated

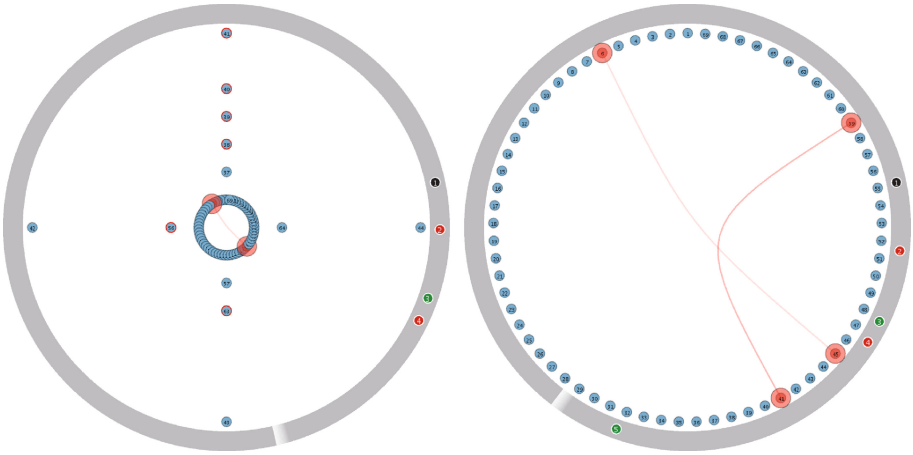

Fig. 6. (a) (Left) A second wormhole attack is detected. Suspicious nodes are highlighted. (b) (Right) The second wormhole link between nodes 41 and 59 is successfully resolved. VA-WAD returns to the monitoring phase.

without taking these two nodes into account. The WAD mechanism also returns to the monitoring phase. It compares the current routing cost of each node with the anchor values. Again, it founds that a new set of nodes advertised lower routing cost than the anchor value.

Figure 6a, depicts the visualization interface after the second alarm. A new message (numbered as 4) has been inserted on the time ring. The *critical node list* now contains the following nodes: 38, 39, 40, 41, 56, and 63. One of them is the source node of the second wormhole tunnel. The mechanism examines their current routing cost, and discovers that node 41 is the malicious one. As shown in Fig. 6b, the wormhole link is resolved, and the network status returns to its normal state. Using this illustration, the user is able to have a full insight of the past network activity at a glance. Even without noticing the past events in real time, the user is highly assisted by the VA-WAD system towards interpreting what preceded in the network. The aforementioned defensive mechanism is able to protect the network integrity from multiple wormhole attacks informing the user about the intruder in an efficient, user-friendly, and engaging way.

4.2 Detection Accuracy

In this subsection, we examine the detection efficacy of the VA-WAD system. The success achieved by a wormhole detection algorithm is measured in terms of the percentage of wormholes detected and the percentage of false positives generated. Table 1 summarizes the detection rate and the false positive rate of VA-WAD. As it can be seen, VA-WAD reports 100 % detection rate in all investigated scenarios. Even in the case of 6 concurrent wormhole links present in the WSN, no attack goes undetected by VA-WAD. The second row of the

Table 1. Wormhole detection as a function of the number of concurrent wormholes

# of concurrent wormhole links	1	2	3	4	5	6
Detection rate (%)	100	100	100	100	99.99	99.91
False positive rate (%)	0.00	0.00	0.00	0.001	0.003	0.008

table depicts the false positive rate achieved by VA-WAD. As it can be seen, the scenarios where a genuine link is falsely reported as a wormhole is negligible.

4.3 Detection Timeliness

Table 2 reports the detection time as a function of the number of concurrent wormhole links in the WSN keeping the number of legitimate nodes fixed and equal to 70. We introduced up to 6 concurrent wormhole links. In the most extreme hostile case, the number of malicious nodes equals to the 8.57 % of the total legitimate nodes of the network. As it can be seen from this table, the WAD algorithm is quick in identifying the wormhole links. As expected, the detection time increases almost linearly with the number of concurrent wormhole links. However, even in the extreme case where 6 wormhole links are present in the WSN, it only requires 7.5 s to detect all wormhole links limiting as such the chance of the malicious nodes to damage the network.

Table 2. Detection time as a function of the number of concurrent wormholes

# of concurrent wormhole links	1	2	3	4	5	6
Detection time (in seconds)	0.4297	1.7547	3.1953	4.725	6.0969	7.5359

Table 3 depicts the obtained results with regard to the detection time as a function of the number of the legitimate sensor nodes. In this scenario, only two wormhole links are present in the WSN. The legitimate number of sensors alters between 50 to 100 with a step of 10. As it can be seen, the proposed algorithm shows quick adaptation to the presented anomalies since the detection is complete within a very limited time. As the number of legitimate nodes increases, the detection time increases as well. This is attributed to the fact that the time required to perform routing cost differentiations lasts more. Moreover, the critical node list is getting bigger, increasing as such the processing time of the WAD mechanism. However, even in the case of 100 nodes, the total detection time is

Table 3. Detection time as a function of the number of sensor nodes

# of legitimate sensor nodes	50	60	70	80	90	100
Detection time (in seconds)	1.2734	1.4406	1.7547	2.0218	2.2205	2.5823

less than 2.6 s. Hence, the recovery process to address a single or more detected wormhole threats is accelerated.

5 Conclusions

The ever-increasing amount of security events reported in mission-critical applications wireless sensor networks are envisaged to support asks for new tools to deal with them. As a novel network security visualization tool, VA-WAD stands out as one such solution. In this work, we proposed a robust, visual-assisted anomaly detection system that is capable of identifying concurrent wormhole attacks; one of the most daunting challenges in the sensor network security field. The VA-WAD system efficiently utilized the routing dynamics in order to monitor and timely detect such attacks. We evaluated the detection accuracy and visual efficacy of the proposed system by simulating a demanding attack scenario, and showed how our tool can be used to expose the attacks and visually correlate the wormhole tunnel. In the future, we intend to validate the VA-WAD system through extended user studies where network analysts and experts will use the system and provide feedback on its usability. Moreover, we will extend the capabilities of the VA-WAD system in order to enable the tool to detect a series of new attack patterns, such as Sybil attacks, Sinkhole attacks, etc.

Acknowledgments. This work was performed within the framework of the Action "Supporting Postdoctoral Researchers" of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund and the Greek State.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks - a survey. Comput. Netw. **38**(4), 393–422 (2002)
2. Chen, X., Makki, K., Yen, K., Pissinou, N.: Sensor network security: a survey. IEEE Commun. Surveys Tuts. **11**(2), 52–73 (2009)
3. Zhou, Y., Fang, Y., Zhang, Y.: Securing wireless sensor networks: a survey. IEEE Commun. Surveys Tuts. **10**(3), 6–28 (2008)
4. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. In: First IEEE International Workshop on Sensor Network Protocols and Applications, pp. 113–127 (2002)
5. Singh, S.K., Singh, M.P., Singh, D.K.: A survey on network security and attack defense mechanism for wireless sensor networks. International Journal of Computer Trends and Technology **11**(2), 1–9 (2011)
6. Sanzgiri, K., Dahill, B., Levine, B., Shields, C., Belding-Royer, E.: A secure routing protocol for ad hoc networks. In: Proceedings of the 10th IEEE International Conference on Network Protocols, 2002, pp. 78–87 (2002)
7. Hu, Y.-C., Perrig, A., Johnson, D.: Packet leashes: a defense against wormhole attacks in wireless networks. In: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, vol. 3, pp. 1976–1986. IEEE Societies (2003)

8. Khabbazian, M., Mercier, H., Bhargava, V.: Severity analysis and countermeasure for the wormhole attack in wireless ad hoc networks. IEEE Trans. Wireless Commun. **8**(2), 736–745 (2009)
9. Papadimitratos, P., Haas, Z.J.: Secure link state routing for mobile ad hoc networks. In: Symposium on Applications and the Internet Workshops, pp. 379–383. IEEE Computer Society (2003)
10. Ban, X., Sarkar, R., Gao, J.: Local connectivity tests to identify wormholes in wireless networks. In: Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ser. MobiHoc, pp. 1–11 (2011)
11. Su, M.-Y.: WARP: a wormhole-avoidance routing protocol by anomaly detection in mobile ad hoc networks. Comput. Secur. **29**(2), 208–224 (2010)
12. Wang, X., Wong, J.: An end-to-end detection of wormhole attack in wireless adhoc networks. In: 31st Annual International Computer Software and Applications Conference, 2007. COMPSAC 2007, vol. 1, pp. 39–48 (2007)
13. Khalil, I., Bagchi, S., Shroff, N.B.: Liteworp: detection and isolation of the wormhole attack in static multihop wireless networks. Comput. Netw. **51**(13), 3750–3772 (2007)
14. Keim, D.A.: Information visualization and visual data mining. IEEE Trans. Visual Comput. Graphics **8**(1), 1–8 (2002)
15. Teoh, S.T., Ma, K.-L., Wu, S.F., Jankun-Kelly, T.J.: Detecting flaws and intruders with visual data analysis. IEEE Comput. Graph. Appl. **24**(5), 27–35 (2004)
16. Thomas, J.J., Cook, K.A.: A visual analytics agenda. IEEE Comput. Graphics Appl. **26**, 10–13 (2006)
17. Awerbuch, B., Curtmola, R., Holmer, D., Rubens, H., Nita-Rotaru, C.: On the survivability of routing protocols in ad hoc wireless networks. In: First International Conference on Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005, pp. 327–338, September 2005
18. Awerbuch, B., Curtmola, R., Holmer, D., Nita-Rotaru, C., Rubens, H.: Odsbr: an on-demand secure byzantine resilient routing protocol for wireless ad hoc networks. ACM Trans. Inf. Syst. Secur. **10**(4), 6:1–6:35 (2008)
19. Wang, W., Bhargava, B.: Visualization of wormholes in sensor networks. In: ACM workshop on Wireless Security, pp. 51–60. ACM Press (2004)
20. Wang, W., Lu, A.: Interactive wormhole detection in large scale wireless networks. In: IEEE Symposium on Visual Analytics Science and Technology, pp. 99–106 (2006)
21. Karapistoli, E., Economides, A.: Wireless sensor network security visualization. In: 2012 4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 850–856, October 2012
22. IEEE 802.15.4^{TM}-2011: IEEE Standard for Local and Metropolitan Area Networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)
23. Draper, G.M., Livnat, Y., Riesenfeld, R.F.: A survey of radial methods for information visualization. IEEE Trans. Vis. Comput. Graph. **15**(5), 759–776 (2009)
24. Diehl, S., Beck, F., Burch, M.: Uncovering strengths and weaknesses of radial visualizations–an empirical approach. IEEE Trans. Vis. Comput. Graph. **16**(6), 935–942 (2010)
25. Brewer, C., Harrower, M.: The Pennsylvania State University. Colorbrewer 2.0 - color advice for cartography. http://colorbrewer2.org
26. Varga, A., Hornig, R.: An overview of the omnet++ simulation environment. In: International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (Simutools), pp. 1–10. ICST (2008)