# SRNET: A Real-time, Cross-based Anomaly Detection and Visualization System for Wireless Sensor Networks

Eirini Karapistoli
Department of Information Systems
University of Macedonia
Egnatias 156, 54006,
Thessaloniki, Greece
ikarapis@uom.gr

Panagiotis Sarigiannidis
Dept. of Informatics and Telecommunications Engineering
University of Western Macedonia
Karamanli & Ligeris Street,
50100, Kozani, Greece
psarigiannidis@uowm.gr

Anastasios A. Economides
Department of Information Systems
University of Macedonia
Egnatias 156, 54006,
Thessaloniki, Greece
economid@uom.gr

## ABSTRACT

Security concerns are a major deterrent in many applications wireless sensor networks are envisaged to support. To date, various security mechanisms have been proposed for these networks dealing with either Medium Access Control (MAC) layer or network layer security issues, or key management problems. Security visualization is the latest weapon that has been added in the arsenal of a security officer who is tasked with detecting network anomalies by analyzing large amounts of audit data. This paper proposes a novel security visualization system for analyzing and detecting complex patterns of sensor network attacks, called SRNET. Both selective forwarding and jamming attacks are identified through visualizing and analyzing network traffic data on multiple coordinated views, namely the multidimensional crossed view, the crossed view perspective, and the track area view. Through simulations, we demonstrate that SRNET is able to help detect and further identify the root cause of the aforementioned sensor network attacks.

## Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General | Security and protection; H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI)—*interaction techniques*

## General Terms

Security Visualization

## Keywords

Wireless Sensor Network Security Visualization, Cross-based Anomaly Detection and Visualization, Detection of Selective Forwarding and Jamming Attacks


*VizSec '13* October 14 2013, Atlanta, GA, USA


## 1. INTRODUCTION

A wireless sensor network (WSN) is a network of cheap and simple processing devices (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. Most applications WSNs are envisaged to support require the remote and unattended operation of a large number of sensor nodes. The unattended nature of the deployed WSNs and the limited resources of the wireless nodes raises immediate administration problems and appoints the security as an increasingly critical element in the network design [1]. Currently, research on providing security solutions for WSNs has primarily focused in three categories [2]; key management, authentication and secure routing, as well as in secure services such as secure localization, aggregation and time synchronization.

All mentioned security protocols are based on particular assumptions about the nature of the attacker. If the attacker is 'weak' (i.e. a passive mote-class attacker), the protocol will achieve its security goal. If the attacker is 'strong' (i.e., a laptop-class attacker), there is a non-negligible probability that the adversary will break in, and will start running some malicious code. Because of their resource constraints, sensor nodes usually cannot deal with very strong adversaries. So, what is needed is a second line of defense: an Intrusion Detection System (IDS) that can detect a third party's attempts of exploiting possible insecurities, and as such, warn for malicious attacks.

An effective IDS must be able to recognize familiar threats as well as identify threats that have not been experienced before. Automated systems are currently very effective in the first of these two categories; that is, a well-designed automatic IDS can very quickly identify a network intrusion if that intrusion has a previously seen pattern of data [3]. Despite the fast development of automated IDSs, the evolving nature of the attacks as well as the scale and complexity of the generated network traffic (also known as the "Big Data" problem) put ever-increasing challenges to the interpretation and understanding of security-related information [4, 5]. Moreover, the state of the art in anomalous activity detection, which is required for recognizing new threats, is very unreliable in automated systems. These systems require vigilant human oversight, 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 an extremely high false positive rate.

In order to meet the inherent analytical needs, the scientific community turned to the Visual Analytics approach. Visual Analytics can be described as "*the science of analytical reasoning facilitated by interactive visual interfaces*" [6]. It is a tight integration of visual and automatic data analysis methods for information exploration and scalable decision support. Whenever automatic systems become insufficient for recognizing malicious patterns, advanced visualization and interaction techniques can be used as a bridge, encouraging the expert user to explore the relevant data and to take advantage of the human perception, intuition, and background knowledge [7, 8]. This feature should be considered as the main benefit of visualization for network security. Moreover, a visual-based approach to the anomaly detection problem does not need a "normal" data set and mainly relies on the superior visual processing capability of the human brain to detect patterns and draw inference. Starting with no prior knowledge of what shape or form the anomalies take, researchers use visualization as a novel tool for discovering the intrinsic properties of normal and abnormal data [9, 10].

This paper contributes to the area of security visualization for wireless sensor networks. We propose a robust *Visual-assisted Intrusion Detection System* (VIDS) devised for the process of detecting complex patterns of abnormal network behavior in large-scale WSNs, called SRNET. SRNET is an integrated system that tackles the sensor network anomaly detection problem by using novel visual analytics technologies. To help address the security visualization challenges, SRNET offers the following contributions;

- A multidimensional crossed view enhanced with a highlight function that monitors the evolving status of two classes of sensor network attacks, namely selective forwarding attacks and jamming attacks.
- A crossed view perspective combined with a track view, which is introduced so as to timely locate the source of the correlated anomaly.
- A novel track area view that tracks the source and the pattern of a potential jamming attack and which enables attribution of the attacker.

The remainder of the paper is organized as follows. In Section 2 we review visualization assisted approaches aimed at detecting network attacks in WSNs. Section 3 introduces the basic features and views of the SRNET visual analytics system. We present experimental results in Section 4. Finally, Section 5 concludes the paper and discusses future extensions.

## 2. RELATED WORK

Several security visualization systems have been proposed in the literature to address the problem of visual-based anomaly detection in WSNs. Early in 2004, Wang and Bhargava [11] 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. MDS-VOW was evaluated for its wormhole detection accuracy through simulations. While efficient, this approach has several deficiencies, especially when applied to dynamic wireless environments (the detection rate drops from over 95% to less than 70% and the ratio of false positives rises from less than 10% to over 80%).

Wang and Lu [12] extended the MDS-VOW concept proposing an improved detection mechanism, called interactive visualization of wormholes (IVoW). IVoW mechanism efficiently integrates automatic intrusion detection algorithms with visual representation and user interaction to support visualization of several wormholes in large scale dynamic WSNs. Simulation results showed that IVoW accelerates the detection process and improves the algorithm accuracy when compared to the MDS-VOW approach.

Wang and Lu [13] proposed an effective approach for monitoring and detecting Sybil attacks in large scale WSNs. This approach uses multiple 2D and 3D views that allow the user to observe the network topology information through multiple aspects and reveal data correlations. Simulation studies showed that the proposed mechanism can effectively identify both direct and indirect Sybil attacks. One drawback of this tool is that greater visualization effort is needed to come up with a firm final resolution.

Recently, Lu *et al.* [14] developed an integrated approach to detect Sybil attacks in mobile WSNs through visualizing and analyzing multiple reordered topology patterns. Different from the previous approach, the automatic reordering and evaluation algorithms used here reveal the malicious nodes in the network topology faster and more accurately. The proposed approach also provides a time-series analysis in order to identify attack durations. This analysis is based on time histograms and an automatic time segmentation method. Overall, this approach was evaluated through a series of real-life attack scenarios, and has shown success at unveiling unknown Sybil attacks.

Abuaitah *et al.* [15], developed a security visualization system, called SecVizer, capable of parsing any QualNet generated traffic trace from both wired and wireless networks. SecVizer combines topology visualization with the parallel coordinate plot technique in order to obtain a faster and more effective detection of network vulnerabilities. By exploring noticeable traffic patterns at both the network topology window and the parallel plot window, the tool has demonstrated its ability to detect various malicious network activities, most notably Distributed Denial of Service (DDoS) attacks. The tool, in its current status, is intuitive enough to allow an analyst to process network events in real-time, but further drill down depth is needed to come up with a firm final resolution.

In the settings of WSNs, Shi *et al.* [16] proposed a novel approach to sensor network anomaly detection and fault diagnosis through visualization. The SAVE system encompasses three distinct visualization components that interpret the topological, correlational and dimensional sensor data dynamics and their anomalies. SAVE was validated through a case study deployment on the real-world large-scale WSN system called GreenOrbs.

As it can be seen, the existing security visualization sys-

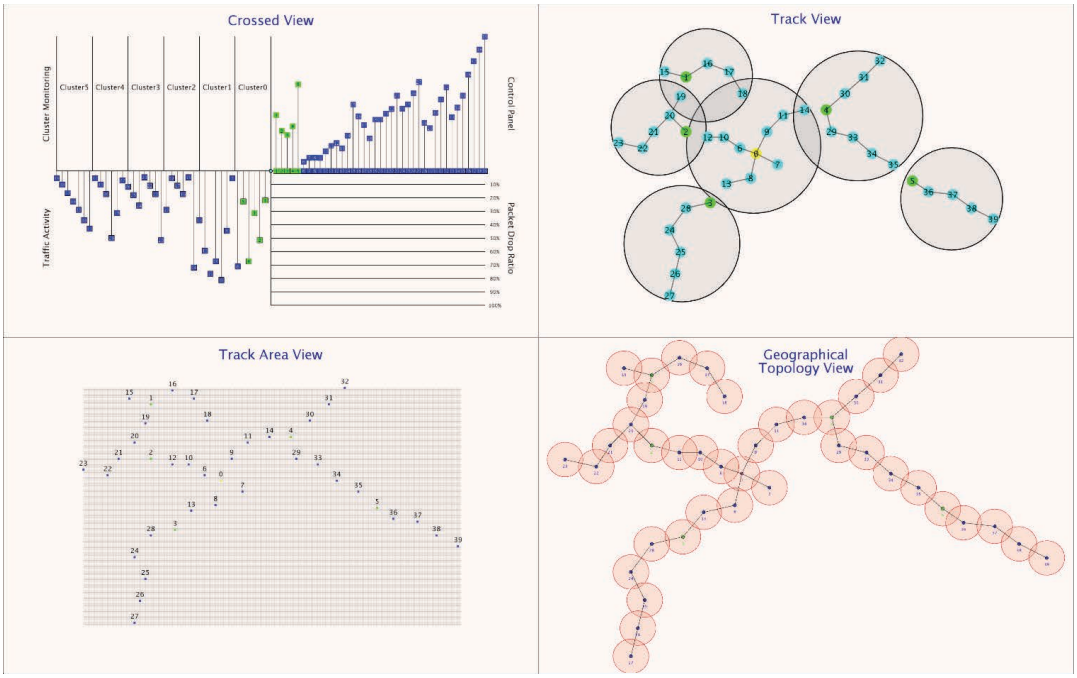


**Figure 1: The SRNET visual interface.**

tems can only deal with a single type of attack instance. Compared to the previous security methods, within SRNET, we apply as well as develop novel views for visually detecting a number of sensor network attacks in one single view.

# 3. THE SRNET SYSTEM

SRNET is a system that fully leverages the power of both visualization and anomaly detection analytics to guide the user to quickly and accurately detect complex patterns of sensor network attacks. At its current state, SRNET detects selective forwarding [17] and jamming attacks [18] against the WSN through visualizing and analyzing network traffic data on multiple coordinated views. Figure 1 shows the whole picture of the provided visualizations of the SRNET system. The main user interface is composed of the Geographical Topology View (lower right), the Crossed View (upper left), the Track View (upper right) and the Track Area View (lower left). In the following subsections, the added value of each of these contributions is discussed in detail.

## 3.1 The Geographical Topology View

The *Geographical Topology View* illustrates the physical topology of the sensor nodes, which resembles a multi-cluster tree structure similar to the one proposed by the IEEE 802.15.4 standard [19]; a dominant communication standard developed to provide low-power and highly reliable wireless connectivity among inexpensive, battery-powered devices. Each node is illustrated with a circle. A different color differentiates the role of each node, which in the case of cluster-based WSNs, can either act as a coordinator (i.e. the principal controller of a cluster) or as a device (i.e. a cluster member with sensing and communicating capabilities). A ring accompanying each node shows the node's transmission range, while the connection lines represent the parent-child relationships between the sensor nodes. While the network under investigation comprises of 40 sensor networks organized into five clusters following the association procedure of the 802.15.4 standard, the SRNET system can support WSNs of arbitrary number of nodes and clusters.

## 3.2 The Crossed View

We devised the concept of the *Crossed View* in order to provide a multidimensional, consolidated, and effective view of the network status to the human behind the monitor. The rationale behind the visualization of the crossed view is the simplicity of the interpretation of the potential threads to the network. The design behind the picture aims at illustrating the thread immediately, without entailing complex actions or complicated human interaction. In order to overcome scalability issues, a similar but more focused view is also provided, which is triggered just with a simple click by the administrator. This focused view is called *Crossed Cluster View* and restricts the view to a simple cluster of the sensor network keeping the provided properties and configuration intact. Figure 2 illustrates the crossed view under normal operation. The whole view is divided into four quarters. All information provided by this view is synchronized. Each quarter endeavors to point a crucial feature of the network. For example, the upper right quarter reorganizes the network topology in a single view, the lower right quarter reveals potential selective forwarding threads together with analytics, the lower left quarter focuses on the jamming effects at each node, and the upper left quarter constitutes the correlated thread analyzer providing a dynamic projection of the state of the whole sensor network.

More specifically, the *Upper Right Quarter* constitutes the control panel of the scheme. It includes all nodes drawn in a square block shape using different colors to distinguish the coordinators from the devices. The placement of the

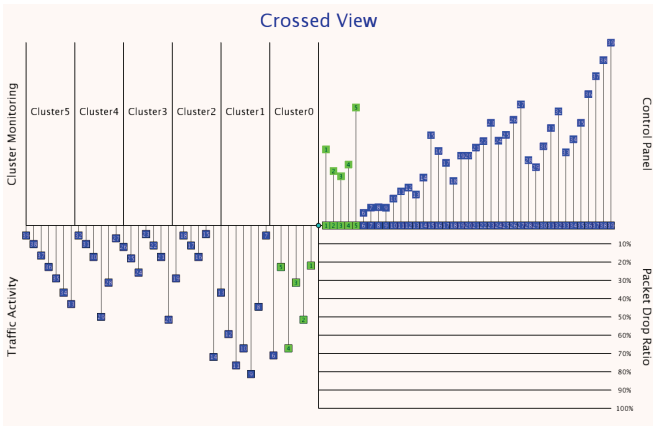


**Figure 2: The Crossed View perspective**

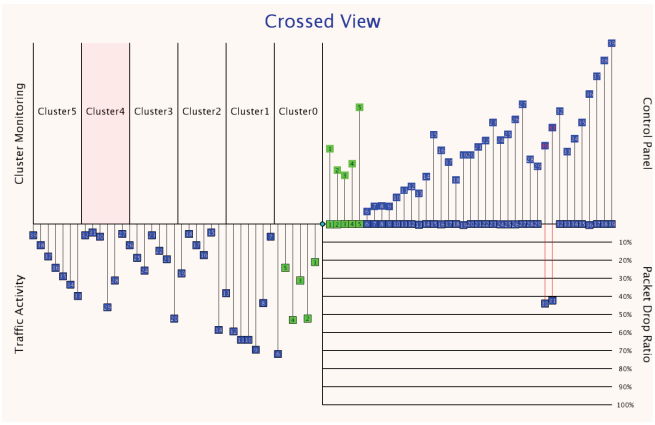


**Figure 3: An example of selective forwarding attack on nodes 30 and 31**

nodes is static and represents the euclidean distance from the sink node (also referred to as the central PAN coordinator). This attribute is designed to offer an immediate image of the location of the attacker, i.e., how far the attack took place from the central point. Hence, the administrator is able to detect a potential thread immediately by taking a look in the upper right quarter of the monitor.

The *Lower Right Quarter* tackles selective forwarding threads. It shows all sensor nodes in square blocks using different colors depending on whether the node is a coordinator or a device. The scale shows the percentage of the dropped packets by each node separately. Conceptually, the illustration is based on an animation. Each block representing the sensor node is progressively moving to the observed value as an animated figure. This method was selected in order to capture the magnitude of the thread instead of a simple value in a time point. Similarly, the animated object is moving backwards when the thread is diminished or resolved so as to highlight the recession of the previous thread. The pace of the animated block depends on the observed window. In essence, a specific time window is maintained for each node representing the percentage of dropped data during this time. The size of the window as well as the frequency of the update define the pace of the animation.

Figure 3 reveals an example of an attack on the nodes 30 and 31. The square blocks representing nodes 30 and 31 are moving towards the bottom edge of the quarter, expressing a considerable drop ratio that should be treated by the observer. The alarm is triggered in the upper right quarter, which plays the role of the analyzer, by changing the color of the nodes' indices under attack. The threshold (drop ratio) triggering this color is considered as a system parameter. Assuming that at least 5% drop ratio is acceptable, under normal network operation, the adopted threshold in the demonstration was set to 5%. The magnitude of the alarm depends on the distance the square block covers. For instance, as the Figure 3 illustrates, currently both nodes experience at least 40% drop ratio. The added value of this perspective lies in the animated graph, where the observer could timely perceive the potential thread, identify the victims, and recognize the level of the thread.

The jamming thread is monitored in the *Lower Left Quarter*. Once more, the animation concept is utilized in order to effectively illustrate potential jamming attacks in the sensor network. All sensor nodes are incessantly moving showing a normal operation. This action represents the ongoing traffic sent rate. In other words, the height where a node is placed within this quarter reveals its traffic sent rate in packets/sec. This rate is normalized between 0 and the maximum rate observed during the network operation. Under normal network operation, all nodes are uninterruptedly moving up and down in accordance with their current network dynamics, i.e., depending on their traffic rates. The rationale behind this coincides with the jamming attack operation. A sensor node that is under jamming attack is unable to send and forward data, hence the number of sent and received data packets tends to zero. To visualize this phenomenon we plot the current traffic rate of each node in order to isolate problematic situations that affect one or more nodes. For example, Figure 4 depicts a jamming attack affecting nodes 16, 17, and 18. It is worth mentioning that in the visualization paradigm of the lower left quarter, the level of detail regarding the traffic rate is deliberately low since the view aims at highlighting the existence of traffic instead of traffic details. Nonetheless, the notification of the thread is also reflected in the upper right quarter, wherein the analyzer takes into account the monitored attack and marks the links of the nodes under (jamming) attack.

One of the major contributions of this work lies in the *Upper Left Quarter*. In this illustration, we tried to encode analytics considering the network cluster's perspective. A set of adjacent columns, one for each existing cluster, dynamically change color introducing the level of granularity of the threads upon each cluster. By highlighting the level of the thread in each cluster separately, we address the robustness and the efficacy of the visualization results. In this manner, the crossed view is applicable to human diagnosis of medium and large scale data, since an administrator is able to determine the granularity and the localization factors, i.e., where the thread is moving and what is the level of the thread. By using the red color as a basis, in order to further highlight the potential attacks, different scaling of thread dimensions are normalized based on the opacity (alpha) color property. Alternatively, a sequential color scheme could be used [20]. We consider the following additive expression for determining the *highlight function* $(HF)^i$, where $i$ stands for the cluster ID:

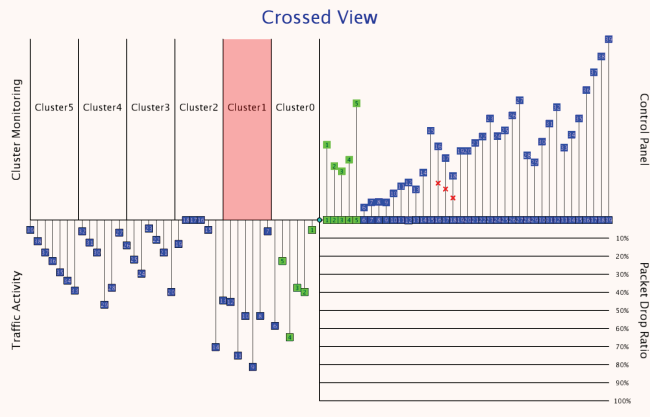


**Figure 4: An example of jamming attack on nodes 16, 17, and 18**

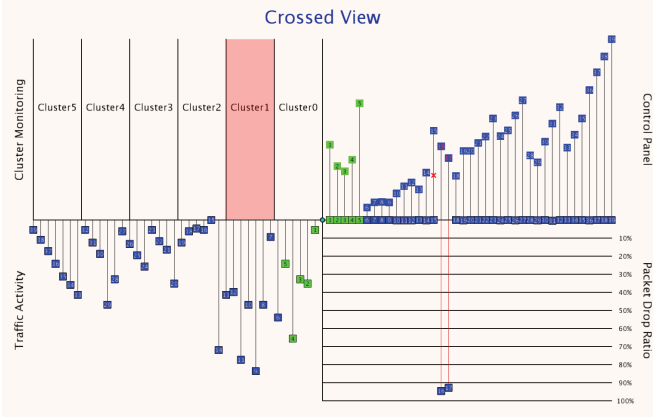


**Figure 5: An example of massive attack on cluster 1 as illustrated by the Crossed View**

$$HF^i = w_{SF} \times SFIM^i + w_J \times JIM^i \quad (1)$$

The parameters $SFIM^i$ and $JIM^i$ denote the impact of the selective forwarding and jamming attack respectively. The weighting factors $w_{SF}$ and $w_J$ provide the relative significance of the two threads to the determination of the highlight function. A wide range of functions may be defined simply by introducing different specifications of the $SFIM^i$ and the $JIM^i$ expressions. However, taking into account that the color opacity takes values in the range [0,255], the highlight function should satisfy the following constraint:

$$0 \leq HF^i \leq 255 \Rightarrow 0 \leq w_{SF} \times SFIM^i + w_J \times JIM^i \leq 255 \quad (2)$$

Functions $SFIM^i$ and $JIM^i$ should be incremental, by showing higher levels of thread by higher values. The former function expresses the level of thread observed by a single or multiple selective forwarding attacks in the sensor network. Intuitively, it should reflect the level of alarm for each cluster separately. Given that the measurement of a node that is under selective forwarding attack is between 0% (no thread) and 100% (complete attack), we define the $SFIM^i$ function as follows:

$$SFIM^i = D^i \times 125 \quad (3)$$

The parameter $D^i$ denotes the average drop ratio of the $i$-th cluster, thus it holds that $0 \leq D^i \leq 1$. If $D^i = 0$, the corresponding cluster experiences no selective forwarding attack. On the contrary, a complete selective forwarding attack occurs in all sensor nodes of a cluster when $D^i = 1$.

On the other hand, the $JIM^i$ function corresponds to the results obtained by a jamming attack to a single or to multiple sensor nodes. In this case, the result is boolean; i.e., a sensor node is or is not under jamming attack. Hence, this phenomenon could be formed based on the number of sensor nodes identified as jamming victims divided by the total number of sensor nodes the cluster includes. Following the above remarks, the $JIM^i$ function is defined as follows:

$$SFIM^i = J^i \times 125 \quad (4)$$

The parameter $J^i$ expresses the portion of the jamming's victim nodes in a single cluster. Obviously, the lowest value the function $J^i$ can receive is 0, meaning that no jamming attacks were observed in cluster $i$, whereas the value 1 dictates a complete jamming attack within the cluster $i$.

As previously mentioned, the weighting factors $w_{SF}$ and $w_J$ express the relative significance of the two threads. The weights offer the capability of dynamically readjusting the point of view regarding the two types of attack. For example, an administrator may consider the jamming attack much more important than the selective forwarding attack since the environment could be outdoor and more hostile to potential jamming attackers. In order to comply with the relation (2), the summation of the two weight factors is one:

$$w_{SF} + w_J = 1 \quad (5)$$

Initially, we treat both threads equally, so it holds:

$$w_{SF} = 0.5, w_J = 0.5 \quad (6)$$

However, a fair treatment of the above weights could be insufficient. For instance, it is important to take into account the frequency an attack is exploited in order to evaluate the attack in the visualization paradigm accordingly. In this way, we further determine the adequate relative significance of the two weights by proposing a dynamic formula that defines the relative values of the above parameters. Let $SFI$ and $JI$ be the number of the selective forwarding and jamming attack instances of the previous time window, $T$, respectively. Then, the weighting factors $w_{SF}$ and $w_J$ are defined as follows:

$$w_{SF} = \frac{SFI}{SFI + JI}, w_J = \frac{JI}{SFI + JI}, \forall SFI + JI \neq 0 \quad (7)$$

One major drawback of the crossed view is the lack of space when the area under investigation is a large-scale WSN. In this case, we tried to address scalability issues by inaugurating a more focused view, similar to the crossed view, called *Crossed Cluster View*. Figure 5 shows a massive attack on cluster 1. Specifically, nodes 16 and 17 experience a selective forwarding attack, while node 15 is under jamming attack. The analyzer in the upper right quarter marks the evidences of the massive attack, i.e., it marks the link of the node 15 and draws the numbers of nodes 16 and 17 with red color. At the same time, the corresponding column in the upper left quarter indicates a massive attack on cluster 1. In

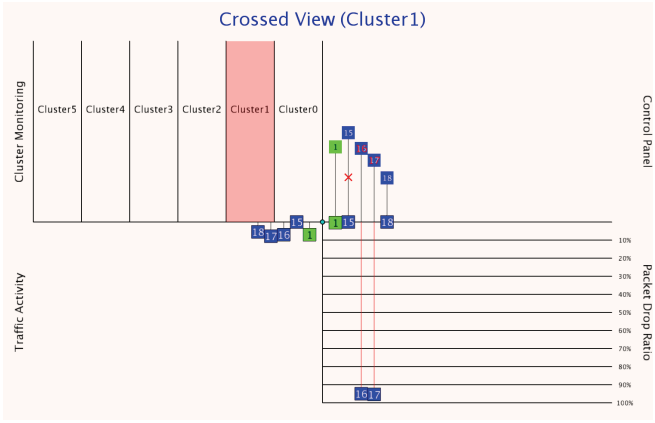


**Figure 6: An alternative view of massive attack on cluster 1 by using the Crossed Cluster View**

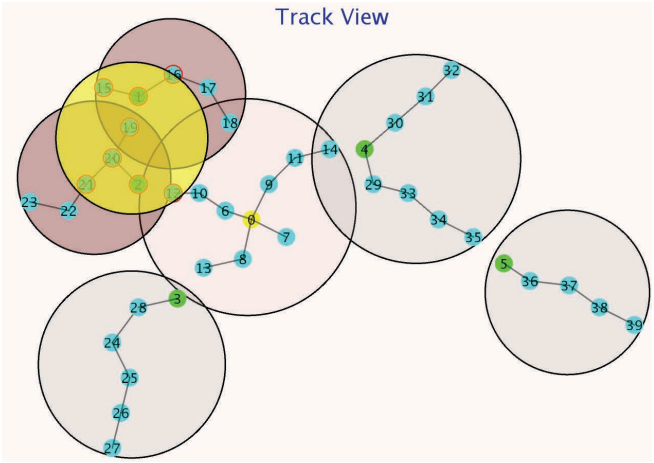


**Figure 7: An example of jamming attack on nodes 1,2, 12, 15, 16, 19, 20, and 21.**

this case, it useful for the administrator to detect significant changes only in the area of interest, i.e., in cluster 1. Hence, by a single click on the related column the user may navigate to the cluster of interest and further investigate the situation. Figure 6 shows the focused view of the crossed cluster view, in which the problematic cluster is isolated allowing for more efficient countermeasures.

## 3.3 The Track View

The Crossed View offers a dynamic projection of the system status in multiple perspectives. It identifies the nodes under attack, the magnitude of the thread, and the dangerous cluster. However, the localization property of the attack is not addressed. The user is aware of the thread but is unaware of two facts; when and where the thread occurred. In order to address this weakness, we further strengthened the capabilities of the SRNET system by introducing the *Track View*. The Track View aims at tracking a potential thread by estimating its source coordinates. Figure 7 depicts the outcome of the Track View as a jamming attack is in progress somewhere between the clusters 1 and 2. In this paradigm, the nodes are designed in circles, having their ID within the circle, while the coordinate nodes have different color from the devices. The area is the normalized version of the original Geographical Topology View, where the connection points implicitly reveal the area of each cluster. Moreover, a black ring distinguishes each cluster to its area. The radius of each ring is calculated so as to enclose all nodes that belong to this cluster. It is worth mentioning that the fill color of each cluster obeys to the rules described in the Section 3.1, i.e., it includes the same color the corresponding column the upper left quarter in the Crossed View has. The rationale behind this is to maintain the thread level to this view, and as such, avoid the disorientation of the observer.

The contribution here lies in the yellow surface that stands between the two clusters. This surface is called the *Track Surface*, and endeavors to estimate the source and the range of the jamming attack. While the notification of the selective forwarding evidences is a trivial issue, i.e., we use a red ring to point a node that experiences a notable drop ratio, as it happens in the case of node 16, the attachment of the jamming source and its properties is a challenge. The *Track Algorithm* describes the logic behind the determination of the jamming activity area. In essence, the algorithm calculates the central point of the surface formed by the coordinates of the nodes under attack. Then, it calculates the maximum distance from this point to the most far cited node; that is, the radius. It is worth mentioning that if there is only a single node under jamming attack, then the Track Surface is simply its range. For example, by observing the Figure 7 we can see the nodes under jamming attack marked with a red ring, namely the nodes 1, 2, 12, 15, 16, 19, 20, and 21. The Track Surface estimates that the intruder may be located in the central point considering all nodes together.

**Algorithm 1** Track Algorithm

**Input:** The coordinates of $z$ nodes under jamming attack $(N = N1_X, N1_Y, N2_X, N2_Y, ...Nz_X, Nz_Y)$.
**Output:** The estimating coordinates of the jamming source $(J_X, J_Y)$ and the its radius $(RADIUS)$.
{ Find the Activity Center }
$tempSumX = 0$
$tempSumY = 0$
**for** each node $i$ under jamming attack **do**
  $tempSumX = tempSumX + Ni_X$
  $tempSumY = tempSumY + Ni_Y$
**end for**
$J_X = tempSumX/z$
$J_Y = tempSumY/z$
$MaxDistanceFromSource = 0$
**for** each node $i$ under jamming attack **do**
  **if** $Euclidean\_Distance(J_X, J_Y, Ni_X, Ni_Y) > MaxDistanceFromSource$ **then**
    $MaxDistanceFromSource = Euclidean\_Distance(J_X, J_Y, Ni_X, Ni_Y)$
  **end if**
**end for**
$RADIUS = MaxDistanceFromSource$

## 3.4 The Track Area View

In order to accurately track the source of a potential jamming attack, we introduce the *Track Area View*. This illustration is complementary to the Track View adding a crucial

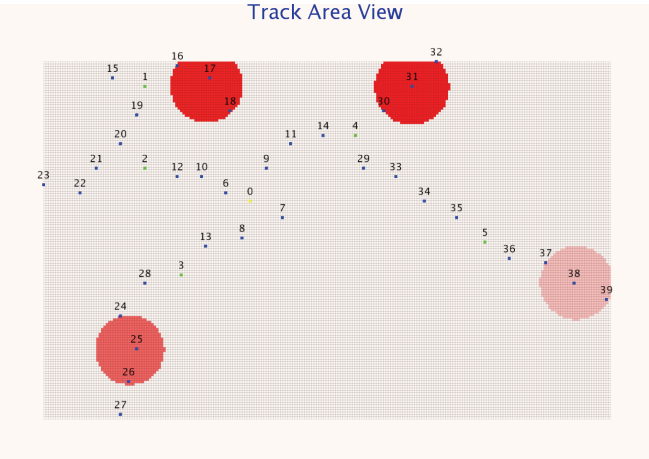


**Figure 8: An instance of four jamming attacks in different time periods.**

value in the dynamic network behavior, namely the time parameter. As previously mentioned, the Track View estimates the location of the intruder as well as the radius of the jamming interference. By applying the track area perspective, we enhance the obtained image with the time dimension. The time correlation is achieved using a different color hue for each monitored jamming attack.

Initially, the whole sensor network is divided into tiles of squared shape. The track area view aims at characterizing the area in accordance with the obtained results using a different color. The red color is used again in order to highlight the dangerous area. The opacity of the color corresponds to the time the attack was perceived. The *Track Area Algorithm (TAA)* describes the way of treating a jamming attack in both distance and time domains. The algorithm consists of two phases, the *Refresh Phase* and the *Update Phase*. The refresh phase is responsible to demonstrate the time dimension by fading the color opacity of the tiles that have been affected by past jamming attacks. It operates periodically so as to pinpoint the time elapsed from previous attacks. The speed of the evaporation, denoted as $T_{refresh}$, determines when the past points of interest will be completely disappeared, and therefore defines the time window of the presence of past attacks. On the other hand, the update phase attributes fresh color to the current, interesting tiles, forming the incomings jamming attack. A fresh color is defined by its opacity; the maximum opacity, i.e., the value of 255, shows a very recent anomaly that needs immediate measures. We employ the simple, but effective, Euclidean distance in order to decide whether a specific tile belongs to the area of the jamming attack. As a result, the fresh colored tiles represent the current time, the weak red areas indicate past attacks, and the white area stands for a clear territory.

Figure 8 captures four jamming attacks occurred in different times. Regarding the first attack within the borders of cluster 5, the color of the Track Surface is quite light. With regard to the attack located in the lower left, the color is light but more intense than the previous attack. The two upper attacks are more recent, while the right one is the current attack. The current attack is drawn in full red in order to highlight an emergent situation.

**Algorithm 2** The Track Area Algorithm

**Input:** .
The number of available tiles ($TI$)
The coordinates of a tile of the given network area ($T_x$,$T_y$)
The coordinates of the estimated jamming source ($S_x$,$S_y$)
The estimated range of the jamming source $S_{range}$
The refresh period $T_{refresh}$
A flag denoting whether there is an active jamming attack (jamming_thread)
**Output:** The new color value $T_{opacity}$.

**Algorithm 3** The Phases of the TAA Algorithm

```
{ Refresh Phase }
for each period T_refresh do
  for each tile TI do
    if T_opacity > 0 then
      T_opacity = T_opacity − 1
    end if
  end for
end for
{ Update Phase }
if jamming_thread == TRUE then
  if Euclidean_Distance(T_X, T_y, S_x, S_y) ≤ S_range then
    T_opacity = 255
  end if
end if
```

## 4. PERFORMANCE EVALUATION

In this Section, we provide an evaluation paradigm of the proposed analytics. In particular, the Track Area Algorithm is assessed so as to demonstrate its accuracy. We consider a bot machine that performs a jamming attack as it moves through the sensor network. The machine follows a fixed path starting from the middle of the network left side and finishing in the middle of the network right side. We assume that the sensor network is placed within an area of $1000 \times 800$ distance points, e.g., meters. We measure the difference of the estimating central point of the attack compared to the actual coordinates of the bot. The bot changes position with a speed of 50 distance points per 60 seconds. The evaluation criterion is the Euclidean distance between the estimated and the actual coordinates of the attack source.

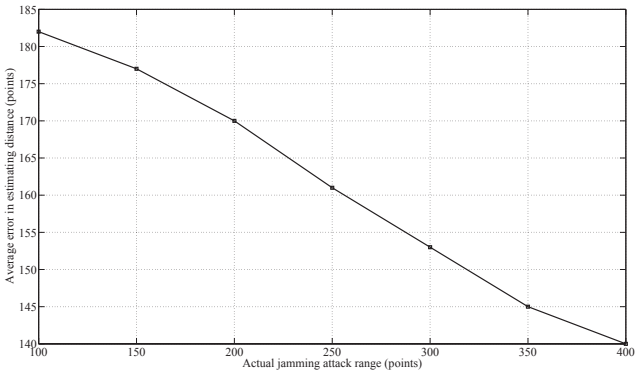


**Figure 9: Average distance error of the Track Area Algorithm in terms of distance units.**

Figure 9 depicts the evaluation results. One important point raised by this figure is that the error level is reduced as the range of the attack comes larger. This is attributed to the fact that a larger attack affects more nodes, so it is more accurate to estimate the coordinates of the source. Another point of interest is that the error is quite limited given that the estimation of the proposed framework is blind. The estimation depends on the nodes under attack, hence, a dense sensor network could provide better estimations compared to a spatial network such as the one under investigation. Nonetheless, the introduced visualization tool is a step toward in the area of visual-based anomaly detection assisting the security professionals at identifying networks attacks.

# 5. CONCLUSIONS AND FUTURE WORK

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, SRNET stands out as one such solution. In this work we proposed a robust, efficient, and effective visualization tool that is capable of identifying selective forwarding and jamming attacks, two of the most daunting challenges in the sensor network security field. The tool is based on multiple point of views, each offering a compelling perspective on addressing potential threads in a sensor network. The add-on values of the presented views focus on identifying multiple threads at a glance, tracking unpredictable jamming attacks, and enhancing the whole view with the time dimension. The accuracy of the tracking capability was assessed and the results indicated that the proposed tool could be a powerful visual analyzer on confronting dynamic, unpredictable, and massive network threads. In the future, we intend to validate the SRNET 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 SRNET system in order to enable the tool to detect a series of new attack patterns, such as Sybil attacks, Sinkhole attacks, Wormhole 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 & Technology), and is co-financed by the European Social Fund and the Greek State.

# 6. REFERENCES

[1] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 3, pp. 6–28, 2008.

[2] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 2, pp. 52–73, 2009.

[3] A. Farooqi and F. Khan, "Intrusion detection systems for wireless sensor networks: A survey," in *Communication and Networking*. Springer, 2009, vol. 56, pp. 234–241.

[4] J. J. Thomas and K. A. Cook, "A Visual Analytics Agenda," *IEEE Computer Graphics and Applications*, vol. 26, pp. 10–13, 2006.

[5] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual data mining," in *Visual Analytics: Scope and Challenges*. Springer-Verlag, 2008, pp. 76–90.

[6] J. J. Thomas and K. A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.

[7] S. T. Teoh, K.-L. Ma, S. F. Wu, and T. J. Jankun-Kelly, "Detecting flaws and intruders with visual data analysis," *IEEE Comput. Graph. Appl.*, vol. 24, no. 5, pp. 27–35, Sep. 2004.

[8] G. Conti, *Security Data Visualization: Graphical Techniques for Network Analysis*, 1st ed. No Starch Press, Oct. 2007.

[9] R. Marty, *Applied Security Visualization*, 1st ed. Pearson Education Inc., 2009.

[10] H. Shiravi, A. Shiravi, and A. Ghorbani, "A Survey of Visualization Systems for Network Security," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 99, pp. 1–19, 2011.

[11] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *ACM workshop on Wireless Security*. ACM Press, 2004, p. 51Ű60.

[12] W. Wang and A. Lu, "Interactive wormhole detection in large scale wireless networks," in *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, nov. 2006, pp. 99–106.

[13] ——, "Visualization assisted detection of Sybil attacks in Wireless Networks," in *Proceedings of the 3rd international workshop on Visualization for computer security*, ser. VizSEC, 2006, pp. 51–60.

[14] A. Lu, W. Wang, A. Dnyate, and X. Hu, "Sybil attack detection through global topology pattern visualization," *Information Visualization*, vol. 10, no. 1, pp. 32–46, 2011.

[15] G. Abuaitah and B. Wang, "SecVizer: A Security Visualization Tool for QualNet-Generated Traffic Traces," in *Proceedings of the 6th Int. Workshop on Visualization for Cyber Security*, 2009, pp. 111–118.

[16] L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, and Z. Su, "SAVE: Sensor anomaly visualization engine," in *IEEE Conference on Visual Analytics Science and Technology (VAST), 2011*, oct. 2011, pp. 201–210.

[17] L. Bysani and A. Turuk, "A survey on selective forwarding attack in wireless sensor networks," in *Devices and Communications (ICDeCom), 2011 International Conference on*, 2011, pp. 1–5.

[18] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the MobiHoc '05 International Symposium*. ACM, 2005, pp. 46–57.

[19] "IEEE 802.15.4™-2011: IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," 2011.

[20] C. Brewer, M. Harrower, and The Pennsylvania State University, "Colorbrewer 2.0 - color advice for cartography." [Online]. Available: http://colorbrewer2.org