Interactive Detection of Network Anomalies via Coordinated Multiple Views

Lane Harrison
ltharri1@uncc.edu

Xianlin Hu
xhu8@uncc.edu

Xiaowei Ying
xying@uncc.edu

Aidong Lu
aidong.lu@uncc.edu

Weichao Wang
weichaowang@uncc.edu

Xintao Wu
xwu@uncc.edu

College of Computing and Informatics
University of North Carolina at Charlotte
Charlotte, NC 28223

ABSTRACT

This paper presents a new approach to intrusion detection that supports the identification and analysis of network anomalies using an interactive coordinated multiple views (CMV) mechanism. A CMV visualization consisting of a node-link diagram, scatterplot, and time histogram is described that allows interactive analysis from different perspectives, as some network anomalies can only be identified through joint features in the provided spaces. Spectral analysis methods are integrated to provide visual cues that allow identification of malicious nodes. An adjacency-based method is developed to generate the time histogram, which allows users to select time ranges in which suspicious activity occurs. Data from Sybil attacks in simulated wireless networks is used as the test bed for the system. The results and discussions demonstrate that intrusion detection can be achieved with a few iterations of CMV exploration. Quantitative results are collected on the accuracy of our approach and comparisons are made to single domain exploration and other high-dimensional projection methods. We believe that this approach can be extended to anomaly detection in general networks, particularly to Internet networks and social networks.

1. INTRODUCTION

Accurate and timely detection of network intrusions is a crucial component for many security and privacy applications. Some intrusions are notoriously difficult to detect due to their complexity and number of variations. Attackers can often easily modify their patterns and signatures to hide from existing detection approaches. Therefore, exploring features of network anomalies during attacks can fundamentally improve various intrusion detection methods.

Several single domain exploration methods for analysis of security data have been developed, as described in our related work in Section 5. Approaches such as these can often overcome the limits of algorithmic methods by integrating interactive exploration with visualizations. However, single domain approaches are not capable of exploring network anomalies across different domains. Such capabilities can be vital in identifying complex intrusions, especially attacks that can be identified only by comparing and correlating several different criteria. The system described in this paper combines node-link diagrams (referred to as the graph domain) with a derived spectral domain. The temporal domain, while crucial to the beginning of an analysis session, is discussed to a lesser extent. Each domain provides a different perspective and can manifest features of the data that the others cannot.

Specifically, this paper presents a CMV approach that facilitates the detection of network anomalies based on changes in network attributes when attacks occur. We concentrate on exploring network features in the graph and spectral domains. The visualization of the graph domain is shown via a node-link diagram, in which node clustering and connectivity can be explored. The visualization of the spectral space represents network nodes as points in a scatterplot, in which anomalous node distributions can be detected through the spectral analysis approaches described in Section 2.1. The temporal domain is visualized as a time histogram of node activity, which allows users to identify and select temporal ranges during which attacks are likely to have occurred. This CMV approach facilitates the exploration of network features using all of the temporal, spectral, and graph domains, thereby providing an effective network anomaly detection solution.

The system described in this paper has been designed to facilitate the detection of Sybil attacks in a time-varying setting. Detection is achieved via the linked visualizations which allows iterative exploration of network features. Because attacks cannot be consistently detected without closely selecting the attack time range, a time histogram is provided to guide users in choosing time ranges in which attacks are likely to have occurred. Several interaction techniques are provided that are designed specifically for facilitating the detection of network anomalies, particularly through exploring subgraphs and examining features of interest in both the graph and spectral spaces. An application of the described approach is demonstrated by detecting Sybil attacks in simulated wireless networks. Neighbor relationships are collected among the wireless nodes during a selected time period and the accumulative results are converted into an adjacency matrix. Then spectral and network graph based analysis techniques are applied to the matrix to create several dimensions that are used in the visualizations. Both case studies and quantitative results are presented to evaluate the effectiveness of the described approach. The CMV exploration ap-

proach is also compared to single domain exploration and other high-dimensional projection methods.

The main contribution of this paper is a CMV exploration approach that facilitates detection of network anomalies based on feature inspection in spectral, graph, and temporal domains. The described system provides suitable visualizations of the graph, spectral, and temporal spaces and interactive exploration mechanisms necessary for detecting anomalies and identifying malicious nodes. Spectral analysis metrics are incorporated into the intrusion detection procedures and visualization, particularly the spectral non-randomness measurement. These metrics are shown to be useful in the intrusion detection process.

The remainder of this paper is organized as follows. In Section 2 we introduce the background of spectral analysis and Sybil attacks. Section 3 presents the CMV exploration approach, including the design of the graph, spectral, and temporal spaces as well as CMV interaction mechanisms. Section 4 describes the experimental results including case studies and quantitative data. Comparative results are also presented between the described approach and other high-dimensional projection and single domain detection methods. We review related work on spectral analysis, Sybil attack detection, and interactive visualization techniques in Section 5. Finally, we conclude and discuss future work in Section 6.

2. NETWORK BACKGROUND

2.1 Spectral Analysis

Topological data is commonly collected in network applications, since it is extremely important for routing. A global network topology records the connectivity relationships among all the wireless nodes. This section describes how spectral analysis can be used to analyze network features in the topology data.

While there are many different mechanisms to describe a network topology, the most straightforward scheme is the adjacency matrix. If there are n nodes in the network, we can construct a $n \times n$ matrix based on the neighboring relationships among them. Generally an entry of '1' implies that the two corresponding nodes are neighbors and '0' if they are not connected. As a special case, we define that a node is not the neighbor of itself, so that true neighbor relationships amongst nodes can be emphasized.

Let $A = (a_{ij})_{n\times n}$ denote the $n \times n$ adjacency matrix of the network. Since we assume that the neighbor relationship is mutual, A is a symmetric matrix and $A = A^T$. Since we assume that a node cannot connect to itself, all the elements on the main diagonal are equal to 0. By the spectral theorem, all eigenvalues of the matrix A are real. Now let us use λ_i to denote the i-th largest eigenvalue of A associated with unit eigenvector $\boldsymbol{x}_i$. Hence, the spectral decomposition of A can be represented as $A = \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \boldsymbol{x}_i^T$. Here we use x_{ij} to denote the j-th element in the vector $\boldsymbol{x}_i$. The value of x_{ij} represents the role of the j-th node in the eigenvector $\boldsymbol{x}_i$.

Suppose the network contains k node communities. Therefore, the largest k eigenvalues and eigenvectors reflect the nodes' association to the k communities. The i-th eigenvalue indicates the density of community i. If node j is a central node in community i, x_{ij} has a large value while $x_{lj} \approx 0$ for $l \neq j$. Figure 1 illustrates a network containing three node communities. Here nodes B, C, and D are the central nodes of the communities and their corresponding elements in the eigenvectors have the largest values.

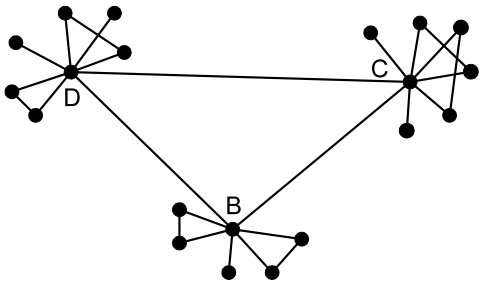


Figure 1: A network with three node communities.

Based on these observations, we can define the non-randomness of node i as follows [31]:

$$R_i = \sum_{j=1}^{k} \lambda_j x_{ji}^2. \tag{1}$$

This metric combines the centrality of the node to the k communities and is weighted by the density of each community. Therefore, the non-randomness measures the node's contribution to the whole network. In Figure 1, the central nodes of the communities have large non-randomness values, while the singleton and noise nodes have small values.

If all the nodes in a network connect to each other independently and randomly, the contribution of nodes to the network topology is directly related to their degree of node connectivity. The more a node connects to the others, the more central role the node plays. Given the degree of connectivity, d_i, we expect that the non-randomness measures of those nodes with the same degree are approximately the same, and those nodes deviated from the majority have a higher probability to be the anomalies. If the node i connects randomly to the rest of the network, its non-randomness value follows the normal distribution whose expectation and variance are upper bounded as follows:

$$\mathbf{E}(R_i) \leq d_i^2 \sum_{j=1}^{k} \frac{\bar{x}_j^2}{\lambda_j} + \frac{d_i}{n}(1 - \frac{d_i}{n}) \sum_{j=1}^{k} \frac{1}{\lambda_j}; \tag{2}$$

$$\mathbf{V}(R_i) \leq \frac{4d_i^3}{n}(1 - \frac{d_i}{n}) \sum_{j=1}^{k} \frac{\bar{x}_j^2}{\lambda_j^2} + \frac{2d_i^2}{n^2}(1 - \frac{d_i}{n})^2 \sum_{j=1}^{k} \frac{1}{\lambda_j^2}, \tag{3}$$

where $\bar{x}_j$ denotes the average value of $\boldsymbol{x}_j$: $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ji}$. Therefore, one reasonable strategy is to label node i as an anomaly if $R_i \geq \mathbf{E}(R_i) + \epsilon\sqrt{\mathbf{V}(R_i)}$. Here the value of ϵ controls the detection sensitivity. For normal distribution, if $\epsilon = 2$, $\mathbf{E}(R_i) + 2\sqrt{\mathbf{V}(R_i)}$ covers more than 95% probability. In practice, we substitute $\mathbf{E}(R_i)$ and $\mathbf{V}(R_i)$ with their upper bounds shown in (2) and (3), respectively.

We are able to make use of these metrics to detect Sybil attacks, which often produce anomalous connectivity patterns when attacking network resources. Features of Sybil attacks are explored in the following section.

2.2 Sybil Attack

The Sybil attack is a particularly harmful attack on wireless networks [5]. This attack has been demonstrated to be detrimental to many important network functions. In a Sybil attack, a single malicious node plays the roles of multiple legitimate members of the network by impersonating their identities or claiming fake IDs.

If there is a group of collusive attackers, each of them can pretend to be the whole group simultaneously at different places in the network, thus manipulating the results of localized voting or data aggregation. Furthermore, Sybil attacks can allow malicious nodes to take control over the entire network by compromising a limited number of physical devices, and defeat the replication mechanisms in distributed systems.

Specifically, when the malicious node sends out network packets with different IDs, the same group of neighbors receive these packets. Therefore, the fake IDs often have many of the same neighbors. When we apply the spectral analysis approach described in Section 2.1 to the scenario, we find that these fake IDs actually form a node community as they move together. Therefore, the non-randomness values of these nodes should be high. On the other hand, if we consider a wireless ad hoc network in which every legitimate node moves randomly and independently in the area, their neighbors change continuously and the non-randomness values are much smaller. Therefore, for legitimate nodes, the average similarity of their neighbors decreases sharply over time, as shown in Figure 2. Conversely, the average similarity of malicious nodes should remain significantly higher over time than that of legitimate nodes. This difference in non-randomness values provides a new metric by which we can distinguish between suspicious and legitimate nodes. Section 4 provides more experiment results.

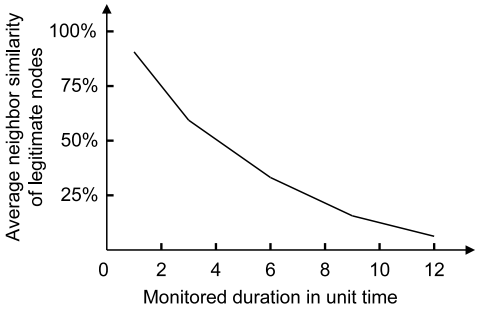


Figure 2: Relationship between the neighbor similarity of the legitimate nodes and the monitored duration.

Based on the previous description, we find that the spectral analysis approach alone cannot guarantee the detection of the Sybil nodes. The challenge is to select a suitable time range for analysis. If we only consider the adjacency matrix at a specific moment, two legitimate nodes in a wireless ad hoc network may be close to each other and thus share many common neighbors. In this case, we cannot distinguish this condition from the cases in which several Sybil IDs spoof the same physical device and move in a coordinated fashion. Therefore, using data from individual time steps may result in high false positive rates. However, when we look at the neighbor relationship for a longer time duration, the neighbors of the legitimate nodes are generally significantly different. To better quantify such changes, we define the ratio between the radio communication range of the wireless nodes and their highest moving speed as one unit time. For example, two nodes C and D are at the same position. If C is static and D is moving in a random direction at the highest speed, after one unit time they will no longer be neighbors. Figure 2 illustrates the changes of the neighbor similarity of legitimate nodes as the monitored duration increases. Over time, legitimate nodes are less likely to have the same neighbors. Based on the results in Figure 2, we choose a minimum detection period around 10 units of time. The adjacency matrices are aggregated during the detection period to calculate the non-randomness values of all the nodes. This value, combined with the connectivity degree of the node, gives us a number of outliers when an attack is occurring. By identifying these clusters and utilizing the interaction techniques in the visualizations, a user can find the malicious nodes and the node they impersonate.

3. COORDINATED MULTIPLE VIEWS EXPLORATION

This section describes the details of our CMV visualization approach. We first present our design of CMV exploration, which visualizes network features from a selected time range in both the graph space and spectral space. A time histogram based on adjacency matrices is then described that suggests possible attack durations through network information. Additionally, we present several necessary CMV interaction methods, which are designed specifically to facilitate the detection of network anomalies.

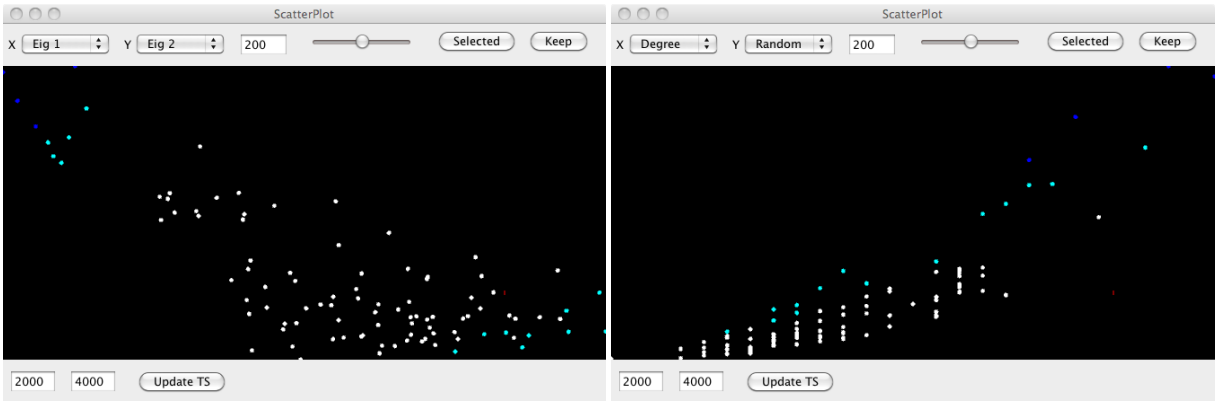
3.1 Visualization Design

The system design objective is to facilitate the detection of anomalous behaviors in networks for a given time range. In order to achieve detection with certainty, we provide visualizations and interactions for the graph, spectral, and temporal spaces. As such, the system makes use of coordinated multiple views by connecting multiple visualizations of the same data and updating all views based on user interaction in any view. What follows is an overview of our system design.

The results of the spectral analysis yields several useful metrics. These results are utilized in a visualization system which provides a workspace where various network features can be analyzed. The spectral analysis yields high-dimensional data, so to facilitate effective interaction we project the data to a 2D sub-spectral space and render them in a scatterplot visualization. For analyzing network features from different spectral dimensions, we allow users to select any two dimensions of the spectral space as the axes of the scatterplot. Figure 3 shows two examples of the spectral space visualization. The image on the left visualizes the first two dimensions of the spectral space, and the right represents the degree and non-randomness metrics. Nodes determined to be suspicious by the spectral analysis methods are colored light blue. For convenience, nodes that are known to be malicious are shown in dark blue.

In particular, we integrate the spectral analysis method which measures the non-randomness features of network node distributions. We have found this metric to be particularly useful when searching for suspicious activity in network data. Visualizing the spectral analysis results allows the user to identify and explore a number of outliers in the network. However, such outliers will often contain both malicious and legitimate nodes. Therefore, the outliers must be examined further, preferably in other domains such as the graph space, to better determine their threat status.

In contrast to the spectral space, network visualization has been studied extensively. As such, we visualize the graph space with a node-link diagram, which visualizes the connectivity relationships of network nodes. Interactions (described in Section 3.2) are well suited to node-link diagrams (as opposed to matrix visualizations). As such, several interaction techniques are provided that allow users to select subgraphs in order to reveal clustering and distribution relationships among the nodes. As Figure 4 shows, our node-link diagram can show isolated subgraphs and highly connected nodes, which are features of Sybil attacks.



(a) Eigenvectors 1 and 2 (X and Y)

(b) Degree and Non-Randomness (X and Y)

Figure 3: Visualization of the spectral space.

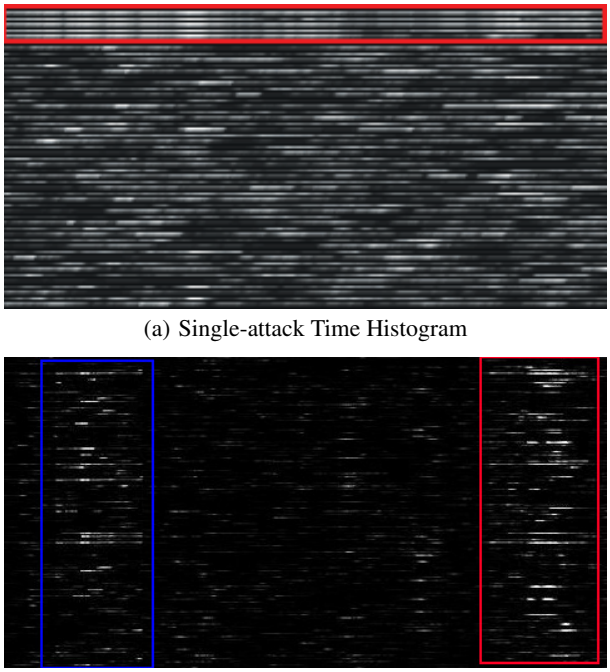
The temporal space is visualized as a time histogram. The time histogram is designed to guide users to select suspicious durations in which the network may be under attack. Effective analysis of temporal information is often crucial to intrusion detection. This problem is relevant to this work because an accurate selection of the attack duration is closely related to the ratio of successfully detected malicious nodes. The purpose of the time histogram is to project our input data as network adjacency matrices from a relatively large time duration (on the order of thousands) to a 2D space that reveals useful visual patterns. This method provides a suitable start for users to identify potential attack durations. An example of areas of interest and user selection is shown in Figure 5. Without the time histogram, users would have to estimate the attack time range, which makes it difficult to accurately detect complex or subtle attacks over a long period of time.

Specifically, we construct a time histogram based on the data from all of the time steps. In the time histogram visualization, the Y axis is evenly divided according to the number of nodes, and the X axis represents time steps (or time divisions if the number of time steps is larger than the number of horizontal pixels on the screen, as explained in the following paragraph). The connectivity degrees of all nodes are mapped to intensity values, with white being highly connected and black representing no connections. To better show regions of interest, a logarithmic scaling factor is applied to the intensity values. Using this view, users can select time ranges of interest by finding areas of increased activity and selecting the corresponding range on the X axis.

If the average number of time steps in the data used is greater than the number of horizontal pixels on the screen, the system collapses time steps evenly in order to show the entire time range on normal-sized computer screens. For example, if the number of time steps is 10,000 and the screen width is approximately 1000 pixels, the system will divide the time range into bins of 10 time steps each and use this data in the time histogram.

3.2 Interaction

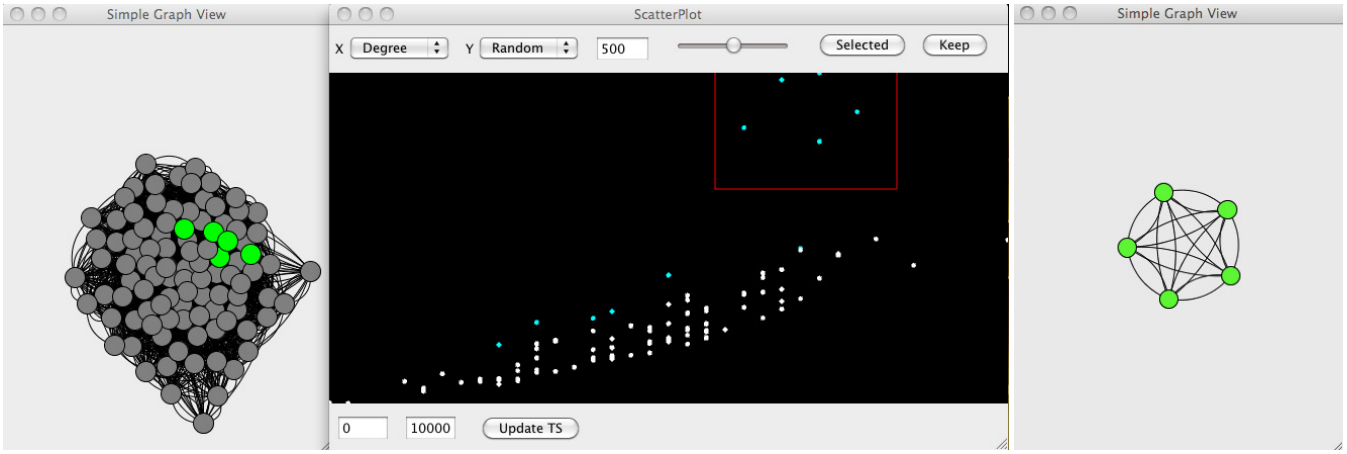
The node-link view provides several basic interactions. Transformation interactions include zooming and panning. Users may change the layout algorithm (force-directed, radial, et cetera) as needed. Selection interactions include box and individual node selection, and all selected nodes are highlighted in the scatterplot



(a) Single-attack Time Histogram

(b) Double-attack Time Histogram

Figure 5: Time Histogram Example: Note the semi-continuous lines at the top of the view in 5(a). In this case, the suspicious activity occurs during the entire duration of the simulation. Seeing this pattern, a user will know to select the entire time range to get more accurate spectral analysis results. Similar results are shown in 5(b), where each area of interest (illustrated as red and blue boxes) indicates an area in which a rise in activity is seen.



(a) Graph before selection

(b) Graph after selection and update

Figure 4: Selecting subgraphs via the spectral space.

view. Showing selected nodes in all views assists the user in determining which outliers are indeed malicious nodes.

Since the spectral space shows the results of non-trivial statistical analysis, we expect that users will usually begin their interaction process there. As such, a larger set of interaction capabilities are provided for this view. Like the node-link view, box and individual selections on the spectral space scatterplot are reflected in the node-link view. To explore scatterplot selections further, the user may temporarily show a new graph containing only the selected nodes. This allows the identification of suspicious sub-graphs and connectivity patterns.

Because the spectral analysis provides several metrics (eigenvectors, non-randomness, degree), our system allows any of them to be placed on the X and Y axes via drop-down menus. This gives the user several possible spaces in which they can find patterns and outliers. Users can adjust the matrix accumulation threshold with the provided interface components. The suspiciousness threshold value from the spectral analysis can also be altered, which increases or decreases the number of automatically-detected outliers. We expect that users will be able to place more confidence in their selections of suspicious nodes by choosing subsets of nodes that have already been determined as suspicious by the spectral analysis.

Dual domain interaction is necessary to allow iterative exploration of hypotheses regarding malicious nodes. Simply identifying outliers is not sufficient, as oftentimes non-malicious nodes are found among the outliers, since some nodes in a network exhibit connectivity patterns different from most other nodes (a mail server, for instance). By allowing users to redraw the graph with only the nodes of interest selected in the scatterplot, users can gather additional information about the nodes that are outliers. For example, since it is known that Sybil nodes tend to communicate frequently [5], they will be shown as a highly connected subgraph whereas benign nodes will often form multiple subgraphs. Users may then make selections in the node-link view identifying the remaining suspicious nodes and continue interaction in the scatterplot view by adjusting the threshold and time ranges to explore the patterns in the refined list of suspicious nodes.

3.3 Detection Solution

The challenges of detecting malicious activity in a given time duration are two-fold. First, a suitable detection strategy must be developed to distinguish between anomalous and normal network features. Second, users are often required to modify potential attack durations for better detection accuracy. Since these two problems are closely related, it is particularly challenging for users to explore solutions for both of them. The described CMV visualization is designed to provide a platform that addresses these issues by giving users the tools needed to analyze complex detection problems.

We use data from Sybil attacks in simulated wireless networks to test our system. Our input data gives an adjacency matrix for each time step. When the user selects a time range, the matrices for the selected time steps are added together to generate an accumulation matrix. A new adjacency matrix is then generated by applying a user-defined threshold value to the accumulation matrix. For example, if the threshold value is 100, then any element in the accumulation matrix that is below 100 will be set to 0, and any element larger than or equal to 100 will be set to 1. This derived adjacency matrix is analyzed using the spectral methods, and the results are reflected in all of the visualization views.

According to the properties of Sybil attacks, we identify the attack features in all three components of our CMV visualization. As we describe in Section 3.1, malicious nodes are often clustered in the graph space. They will also be classified as outliers under the spectral non-randomness measurement. Oftentimes, they can be initially identified by bright, continuous line segments in the time histogram. However, selections in the time histogram must be examined further in the other views to confirm malicious activity.

Based on the network features in the components of our CMV visualization, we develop an effective mechanism to locate the Sybil nodes. Specifically, users start from the time histogram view to select a suspicious time duration with the illustrated line patterns. The system accumulates the network adjacency matrices within this duration and produces a (0, 1)-matrix as described above. After the spectral analysis is performed, both the graph space and spectral space are visualized automatically according to the selected time duration. Then users can utilize the provided interactive exploration methods to search for related attack features in both spaces.

The advantage of our approach is as follows. Once a malicious node is identified, due to the group nature of Sybil identities, we can quickly locate other "partner" nodes by visualizing the highly connected clusters in the graph space. We can remove all the malicious nodes from the network and repeat our detection strategy, until the network is clear. A network can be identified as clear if all the outliers in the spectral space do not appear in the same clusters in the graph space.

It is worth mentioning that the same pre-determined attack duration will not satisfy the detection requirements under different conditions. For example, if the selected time duration is too long, the attack features will be hidden under the noise in the network. On the other hand, if the duration is too short, we will not be able to distinguish Sybil identities from legitimate nodes that happen to move as a group. We experiment with different values of the duration and the results are presented in Section 4.

Our prototype system is implemented using the Java language and a combination of JUNG (Java Universal Network / Graph Framework)[9], JOGL (Java bindings for OpenGL)[23], and MATLAB[8]. JUNG is an open-source library for displaying node-link diagrams. The matlabcontrol library[10] is used to make calls to MATLAB from Java throughout the interaction process to allow convenient and timely calculations for the spectral analysis.

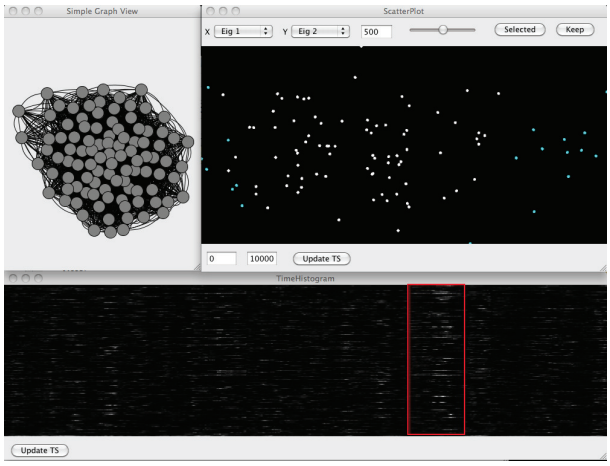
4. RESULTS AND DISCUSSIONS

In this section, we first describe two case studies that demonstrate how our CMV exploration approach can be used for detecting network anomalies. We then present our simulation results to evaluate the effectiveness of our approach. We also compare our CMV exploration approach to other single domain methods and high-dimensional projection mechanisms such as MDS and PCA.

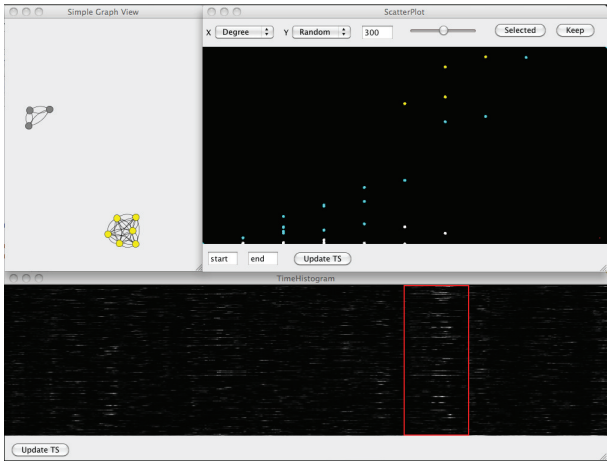
4.1 Case Studies

Two case studies are provided to demonstrate the robustness of our approach. In the first example, malicious nodes that attack extensively in a short time duration are identified. In the second example, we show the need for the CMV exploration by studying a more subtle variant of Sybil attacks. Both studies demonstrate that the described CMV exploration can be used to detect malicious nodes that are difficult to locate when only using single domain detection methods.

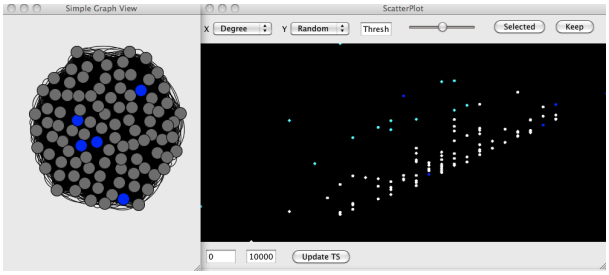
In the first case study, the goal is to identify Sybil nodes that are attached to the same physical device. Initially, our system provides the visualization of the entire time duration, as shown in Figure 6(a). The user may select some of the seemingly clustered points at the right of the scatterplot and re-calculate the node-link diagram, but will find that these form several separate subgraphs, which is not indicative of attacking nodes. Therefore, the user must further leverage the interactive capabilities of the system further to find suspicious activity. The first step is to select a suitable time duration. In the time histogram, the user notices several areas in which multiple nodes have increased activity that is sustained over a certain time period (that is, a range along the X axis) as shown in Figure 6(a). By selecting these time ranges and examining nodes in the spectral and graph spaces for related clusters, the user will be able to determine which nodes exhibit patterns of an attack. Specifically, when the user reaches the time range indicated by the red box in the time histogram of Figure 6(a), the user is shown the updated graph and spectral space visualizations. The user then selects



(a) Initial view of all timesteps. The red box in the time histogram represents a user selection of a time range. The results of this selection are shown in the following figure.



(b) Results after the selecting an appropriate time duration, adjusting accumulation thresholds, and creating a subgraph of suspicious nodes from a selection of outliers in the scatterplot.



(c) Results for the entire time duration. Malicious nodes are shown in dark blue for reference.

Figure 6: Snapshots from the first case study.

the degree and non-randomness measures for the X and Y axes, respectively. This results in a number of nodes being automatically labeled as suspicious (colored light blue).

After adjusting the threshold to examine how the outliers change, the result in the top right of Figure 6(b) is shown. By selecting the nodes with the largest non-randomness values, the user reduces the node-link diagram to a small number of nodes. It is immediately obvious that of the outliers shown in the graph view, some are highly connected and some are not. If the highly connected subgraph does not have such a regular structure, the user can further refine her/his selections through interacting with the spectral space visualization. At this point, by finding several nodes clustered in both the spectral and node-link spaces, all malicious nodes in the selected time range are correctly identified. The final detection result is shown in Figure 6(c). Comparing Figures 6(a) and 6(c), we can see that this Sybil attack does not show an obvious pattern in the initial view. Rather, it can only be detected by the related features from both views. After the nodes responsible for the first attack have been identified, the user can take the list of malicious nodes and see if they are responsible for the second attack by comparing their identification attributes to the results of performing similar interactions on the second suspicious time range.

In the second case study, the user must identify the nodes responsible for a different style of Sybil attack. An initial visualization for the entire time duration is shown in Figure 7. This case is interesting, since the view that contributes the most to a successful identification is the node-link diagram, and the spectral space visualization is used to verify the results. Via the time histogram, the user will notice that there is a sustained activity rate in some nodes across the entire network. It should be noted that sustained activity does not always imply malicious activity, but only suspicious activity. For example, a highly visited web server can generate high activity throughout the network lifetime.

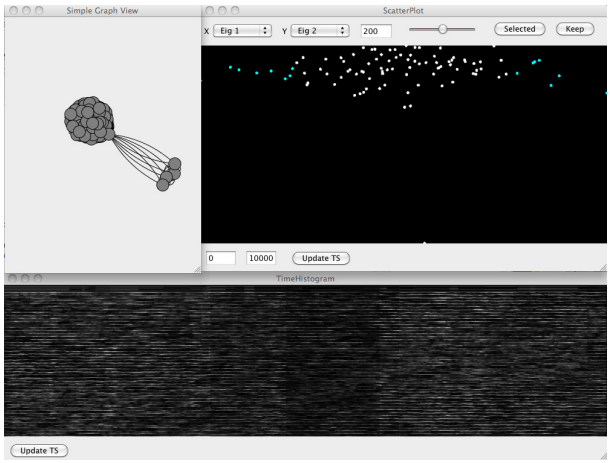
Since this particular attack occurs throughout the entire dataset, the initial view that shows the entire time range will show outliers in the node-link view. This is demonstrated in 7(a). The user will confirm this by selecting the high activity portion of the time histogram, which happens to span most of the temporal space. After adjusting the threshold values higher to accommodate a larger time range, the user will observe that the outlying nodes are communicating to only one node. The user can then select these nodes in the graph space visualization and see where they lie in the spectral space. As shown in Figure 7(b), it turns out that all the nodes are classified as the same point when the user selects eigenvectors 1 and 2 as their respective X and Y axes in the spectral space. The user can conclude that this is highly suspicious activity and mark the outlier nodes and their proxy for isolation and investigation. Comparing the locations of malicious nodes from Figures 7(a) and 7(c), we can see that they are hidden in the original spectral space without the information from the graph space.

4.2 Quantitative Results

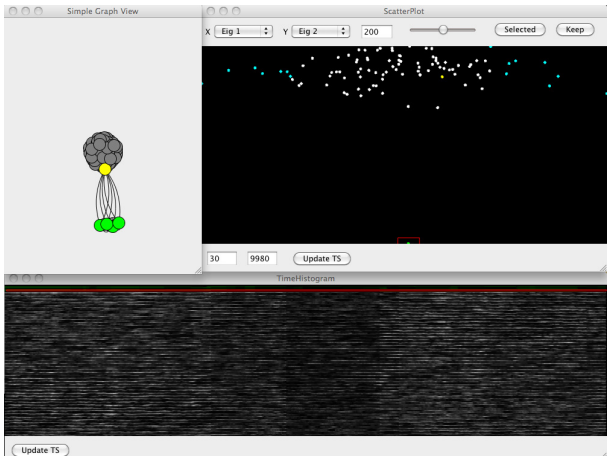
This subsection presents the simulation results of the described approach. We first introduce the configuration of the simulated network. The detection accuracy of the proposed approach and its relationship to the parameters of the mechanisms are then presented.

Simulation Setup

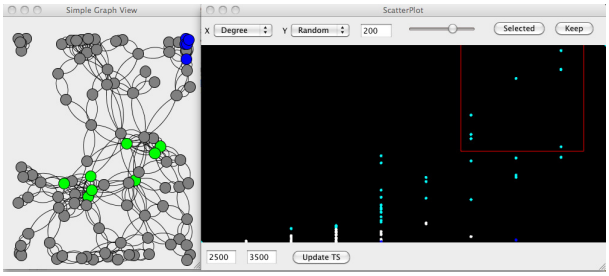
The experiments are conducted in two phases. In the first phase, we simulate the network topology changes of a wireless ad hoc net-



(a) Initial view of all timesteps.



(b) Results after confirming the time range on the the time histogram and selection of the outliers in the graph view. The proxy is selected in the graph space (yellow) and the Sybil nodes are selected through the outlier in the scatterplot (green).



(c) Showing the locations of the malicious nodes (dark blue) and an incorrect selection of outliers (green). This demonstrates that a correct temporal range selection is crucial.

Figure 7: Snapshots from the second case study.

work that are caused by node movement. In the second phase, our detection approach is tested on detecting Sybil attacks and locating the fake identities. The mobile nodes are deployed in a square area with the size of $1400 \times 1400m^2$. The radio communication range r of the wireless nodes is set as $250m$, and any two nodes having a distance shorter than r can directly communicate to each other.

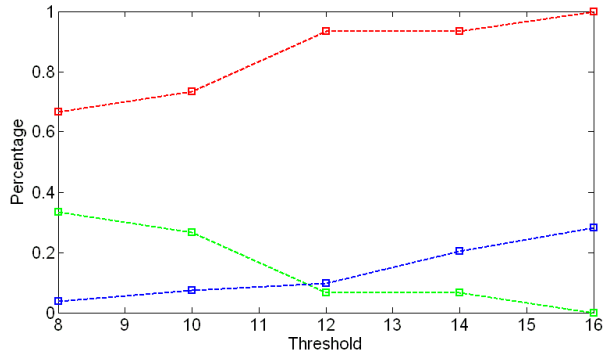
Within the simulated area, 100 nodes are randomly and uniformly distributed. We adopt the random trip movement model proposed in [2] to describe the moving patterns of the nodes. We assume that the highest moving speed of the nodes is 5 m/s. In every simulation, we sample the network topology at the interval of $r/highest\ speed$. Totally 200 samples of the network topology are collected during the simulation period. Within the network, we randomly choose one to five nodes to act as the attacker. Each malicious node can interact with other nodes using two identities simultaneously. We assume that the legitimate nodes cannot identify those fake identities by examining the properties of the radio signals. We generate twenty-two initial node deployments in the network. For each node deployment, we produce ten different node movement patterns.

The detection accuracy of the proposed approach is evaluated by the false alarm rate. Specifically, we consider two types of rates: false positive and false negative. In a false positive mistake, a legitimate node is incorrectly identified as an attacker. In a false negative alarm, a Sybil node is incorrectly identified as a legitimate user. The simulation results show that some parameters of the proposed approach have opposite impacts on the two false alarm rates and some tradeoff must be carefully assessed.

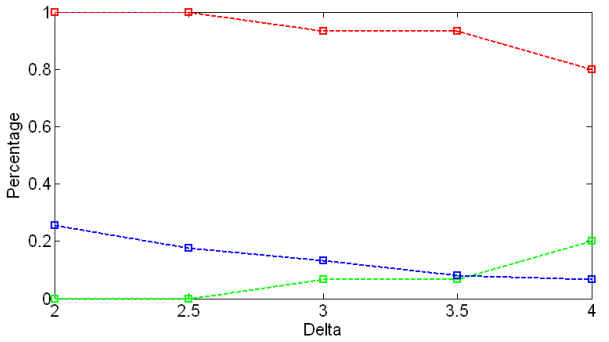
Simulation Results

Figure 8 illustrates the detection results when we apply only the spectral mechanism. Here we investigate the relationship between the detection accuracy and the parameters of the spectral method. We are interested in the impact of two parameters: $threshold$ and $delta$. Specifically, we use the average results of three selected attack datasets for each value of $threshold$ and $delta$ shown in Figure 8. Since the adopted spectral mechanism can analyze only (0,1)-integer matrices, the first parameter $threshold$ is used to convert the accumulated adjacency matrix into a (0,1) matrix by comparing the connectivity counts of the nodes to the selected threshold value. We can see that the selected threshold value will directly impact the detection results. As the threshold increases, the Sybil nodes stand out because of the consistency of the connections among the fake identities. This leads to a lower false negative rate. However, as the threshold increases, the expectation value $E()$ in Equation 2 starts to decrease and more legitimate nodes with a large non-randomness value fall into the suspicious area. Therefore, the false positive rate increases as well. Figure 8.(a) shows the relationship between the false alarm rates and the selected threshold value, which is measured by the number of unit time as defined in Section 2. Here the parameter $delta$ has the fixed value 5. We can see that the changes of the curves match our analysis.

Similar analysis can be applied to the choice of the parameter $delta$. The parameter threshold is fixed as 12. As the value of $delta$ increases, we define a wider range of the non-randomness value for the legitimate nodes. Therefore, fewer legitimate nodes will be incorrectly labeled as attackers and we will have a lower false positive rate. However, the wider range of the non-randomness values will make it more difficult for us to distinguish the Sybil nodes from the legitimate identities and lead to a higher false negative rate. The results in Figure 8.(b) demonstrate such changes.



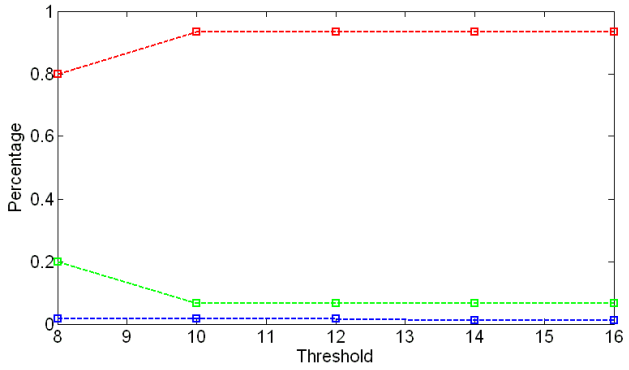
(a) false alarm rate *vs* threshold value



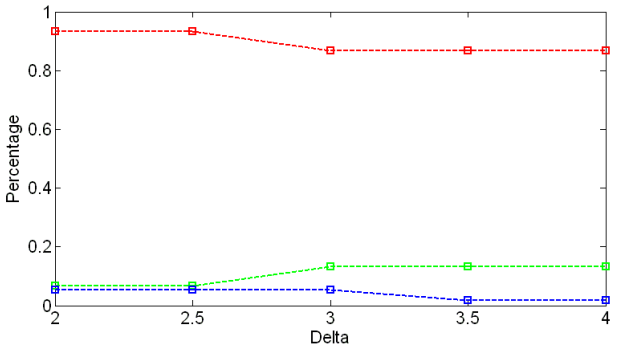
(b) false alarm rate *vs* delta value

Figure 8: Detection accuracy of the spectral approach. The red line indicates the percentage of the malicious nodes that are detected; the green line indicates the percentage of the malicious nodes that are not detected; and the blue line indicates the percentage of legitimate nodes that are incorrectly labeled as Sybil nodes.

The results in Figure 8 show that the spectral analysis approach alone can not provide sufficiently accurate detection capabilities. Figure 9 illustrates the simulation results of our CMV exploration approach. For a fair comparison between the single spectral domain approach and our CMV approach, we use the same parameter delta in Figure 9(a) and parameter threshold in Figure 9(b) as those in Figure 8. The only difference is that a wireless node in our approach is labeled as a Sybil identity only when it demonstrates anomaly in both domains, specifically the topology value of this node is above the threshold and it is located above the delta specified non-randomness curve. Here a wireless node is labeled as a Sybil identity only when it demonstrates anomaly in both domains. The results in Figure 8 and 9 demonstrate that our CMV exploration approach can help reduce both false positive and false negative mistakes. The percentages of the missed malicious nodes are reduced from [0, 0.35] to [0, 0.2] and the percentages of legitimate nodes that are incorrectly labeled as Sybil nodes are reduced from [0.05, 0.3] to [0.02, 0.08].



(a) false alarm rate *vs* threshold value



(b) false alarm rate *vs* delta value

Figure 9: Detection accuracy of the CMV approach.

4.3 Discussion

While there are several different mechanisms that can be used to calculate the primary components of a matrix, we choose the spectral approach over other schemes because of its simplicity and consistency in visual representation. For example, the multi-dimensional scaling (MDS) mechanism [25] can reconstruct the relative positions of the nodes based on their neighbor relationships. However, the reconstructed result can rotate freely before the positions of several anchor nodes are determined. To better justify our decision, we compare the spectral analysis results with the results of principal component analysis (PCA) [28] and MDS. In Figure 10, we can see in all three schemes the attackers are outliers. However in the spectral space, the malicious nodes are always at the upper-right corner in the space, since they have large non-randomness values. Therefore, users may consistently focus their attention on these regions during the interactive detection process and disregard entities below the decision line in the spectral space. In contrast, PCA and MDS produce an inconsistent positioning of clusters, so users must explore all the different clusters to find malicious nodes.

Performance for the described system can be discussed in terms of visualization and spectral scalability. Since the most detailed analyses are performed in the scatterplot and node-link diagrams, these are of particular interest in terms of performance.

The limitation of the node-link approach is that it can only visualize several hundred nodes effectively on a normal sized display. To address this limitation, the proposed system can easily make use of more effective space filling algorithms, or could implement an administrator-defined hierarchical schema based on the network being observed, along with the necessary interactions to handle hierarchical constructs. Scatterplots are also limited in terms of screen space. However, since users will usually filter out nodes classified as normal by the spectral analysis, this is only a problem if the number of outliers is in the thousands when the user has a normal sized display.

Our detection solution and case studies have demonstrated that Sybil attacks cannot be successfully detected with a high degree of certainty by using the graph space or spectral space visualizations alone. However, by facilitating the detection of attack features in the graph, spectral, and temporal domain, users may isolate malicious nodes in a timely and accurate manner.

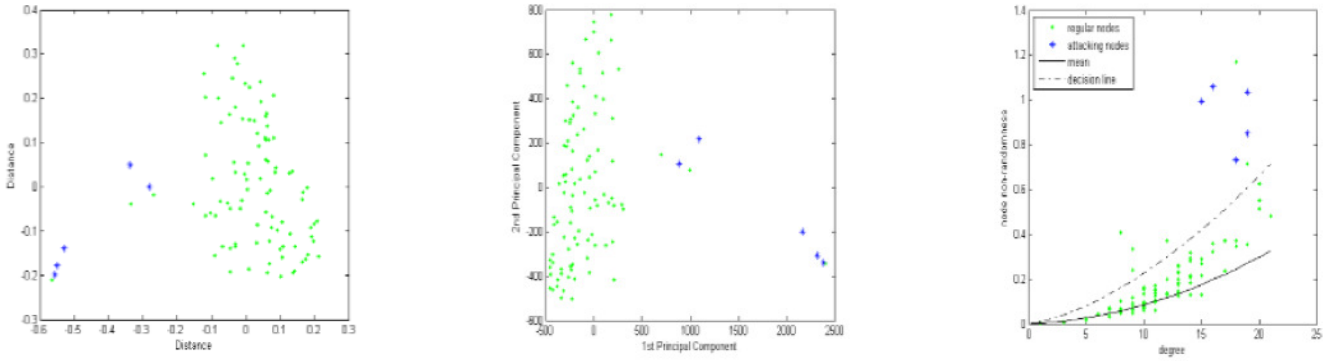
5. RELATED WORK

Interactive detection of intrusion attacks is a popular topic in security visualization. For clarity, we only review the most relevant work on spectral approaches for analyzing network properties, Sybil attack detection methods, and interactive networking detection and CMV exploration methods.

5.1 Spectral Approaches

Spectral methods are a part of graph theory, and have been shown useful in applied mathematics and scientific computing. However, to the best of our knowledge, there have not been many spectral approaches in visualization. The most related work is an introduction of spectral methods by Seary and Richards, who applied spectral methods to discover cohesive clusters and localized features of a network [21]. This paper is the first spectral visualization approach to secure a wireless network.

In the field of network analysis, researchers have explored spectral methods to describe network properties and their relationships. For example, Ying and Wu [30] proposed a spectral property preserving mechanism to study important topological features of network data. Their approach could better study the general properties of the social networks. Wang *et al.* [26] proposed a graph theoretical approach with diffusion and spectral methods based on their previous evidence graph model. Their graph spectral methods could identify crucial elements and patterns of attack by extracting the important graph structure. Our CMV exploration approach provides a platform to integrate suitable spectral analysis methods for intrusion detections.



(a) (b) (c)

Figure 10: Comparison among the mechanisms of (a) MDS, (b) PCA, and (c) spectral. The blue stars are Sybil nodes and the green dots are legitimate ones.

5.2 Sybil Attack Detection in Network

Sybil attack is a harmful attack on distributed systems and wireless networks [5]. Newsome *et al.* have systematically classified these attacks into several types and analyzed their threats to wireless sensor networks [17]. The following provides a brief survey of Sybil attack detection methods in the field of security. In contrast to these methods, our approach does not require any special devices or hard assumptions on the network scenarios.

Based on the detection mechanisms, we divide the previous approaches into three categories: identity, location, and signal-print based methods. Identity-based approaches usually mitigate the Sybil attacks by limiting the generation of valid node information, such as the pre-distributed secret keys [17]. For example, a detection approach was proposed for vehicular ad hoc networks through possible explanations for collected data of each node [7].

Location-based approaches utilize the fact that each node can only be at one position at a specific moment. Localization algorithms, such as SeRLoc [14], were proposed to allow sensors to determine their locations under known attacks including Sybil attack. The geometric properties of message transmission delay were also explored to reduce the impacts of Sybil attacks [1]. In [19], every node signed its ID and position and sent this information out in several random directions. The different positions signed by multiple replications of the same node had a high chance of being detected.

In the signal-print based detection mechanisms, the investigators attempt to collect the properties of the radio signals and detect the false claims of the node identities. For example, in [6], multiple access points measured the signal strength from a node to form the signalprint and used it to detect Sybil nodes. A similar idea was adopted in [4]. The approach in [29] integrated a series of position claims and witness reports in VANETs to detect Sybil nodes. In [3], the radio signal transient shape at the start of a packet was used to identify a physical node and detect Sybil nodes.

5.3 Interactive Visualization Techniques

Interactive techniques are important and necessary for effective data exploration in visualization. These techniques are often used in coordinated multiple views (CMV) visualizations. By interacting with the visual features, analysts can gain insight into the data. At the same time, through interaction, it is possible for analysts to correlate features from multiple perspectives [11]. For example, the XmdvTool [27] is a system that combines several visualization methods with interactions. Users can explore their data in a variety of formats and views. Zhao *et al.* [32] combined node-link diagrams [20] with Treemaps [22] interactively to benefit from the efficiency of Treemaps and structural clarity of node-link diagrams. Nathalie *et al.* [16] built the system MatrixExplorer to explore social networks by using a CMV approach consisting of matrices and node-link diagrams. They provided several functions for users to interact with matrices, while the node-link diagram views provide additional information on social network features. Kosara *et al.* introduce the time histogram as a means of visualizing large time-varying data [12].

Several mechanisms have been designed to explore network data and its security properties. For example, in [24] BGP protocol data were used to characterize routing behaviors. Network and port scan attacks were studied in [13, 15]. Interactive visualization can also help explore complicated data structures such as attack graphs [18]. We combine the graph, spectral, and temporal spaces in a visualization to analyze network anomalies through iterative exploration.

6. CONCLUSION AND FUTURE WORK

This paper presents a CMV approach that facilitates the detection of Sybil attacks of varying time durations in simulated wireless networks. The described system allows users to detect suspicious activity through the temporal space and analyze network anomalies via visual features on both graph space and spectral space, thus providing new exploration capabilities by integrating network features from different perspectives. We have designed CMV visualization with essential interactive exploration methods for studying the network features required to detect malicious activity. The given results and discussion indicate that our approach provides a suitable detection mechanism that has potential to be extended to general network security applications or social network analysis.

For future work, we first plan to test the described system with network flow data. We are also interested in utilizing existing spectral analysis approaches to explore other network features that can be affected during attacks. Furthermore, we plan to test our prototype system with other types of attacks in wireless networks with the goal of achieving a generic intrusion detection visualization system for wireless networks.

7. ACKNOWLEDGEMENTS

This research is supported by DOE Award No. DE-FG02-06ER25733 and NSF Awards Nos. 0754592, 0831204, and 1047621. This material is based upon work supported by the U.S. Department of Homeland Security under Award Number: 2008-ST-104-000017.

8. DISCLAIMER

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

9. REFERENCES

[1] R. A. Bazzi and G. Konjevod. On the establishment of distinct identities in overlay networks. In *Proceedings of ACM PODC*, pages 312–320, 2005.

[2] J.-Y. L. Boudec and M. Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *Proc. of IEEE Infocom*, 2005.

[3] B. Danev and S. Capkun. Transient-based identification of wireless sensor nodes. In *Proc. of IPSN*, pages 25–36, 2009.

[4] M. Demirbas and Y. Song. An rssi-based scheme for sybil attack detection in wireless sensor networks. In *Proceedings of WoWMoM*, pages 564–570, 2006.

[5] J. R. Douceur. The sybil attack. In *the First International Workshop on Peer-to-Peer Systems*, pages 251–260, 2002.

[6] D. B. Faria and D. R. Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of ACM WiSe*, pages 43–52, 2006.

[7] P. Golle, D. Greene, and J. Staddon. Detecting and correcting malicious data in vanets. In *Proc. ACM international workshop on Vehicular ad hoc networks*, pages 29–37, 2004.

[8] T. M. Inc. Matlab. `http://www.mathworks.com/`.

[9] T. N. Joshua O'Madadhain, Danyel Fisher. Jung: Java universal network/graph framework. `http://jung.sourceforge.net/index.html`.

[10] J. Kaplan. matlabcontrol: A java api to control and interact with matlab. `http://code.google.com/p/matlabcontrol/`.

[11] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8:1–8, 2002.

[12] R. Kosara, F. Bendix, and H. Hauser. Timehistograms for large, time-dependent data. In *VisSym*, pages 45–54, 340, 2004.

[13] K. Lakkaraju, W. Yurcik, and A. J. Lee. Nvisionip: netflow visualizations of system state for security situational awareness. In *Proceedings of ACM workshop on Visualization and data mining for computer security*, pages 65–72, 2004.

[14] L. Lazos and R. Poovendran. Serloc: Robust localization for wireless sensor networks. *ACM Trans. Sen. Netw.*, 1(1):73–100, 2005.

[15] C. Muelder, K.-L. Ma, and T. Bartoletti. Interactive visualization for network and port scan detection. In *RAID*, 2005.

[16] H. Nathalie and J.-D. Fekete. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):677–684, 2006.

[17] J. Newsome, R. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: Analysis and defenses. In *Proceedings of IEEE IPSN*, pages 259–268, 2004.

[18] S. Noel and S. Jajodia. Managing attack graph complexity through visual hierarchical aggregation. In *Proceedings of ACM workshop on Visualization and data mining for computer security*, pages 109–118, 2004.

[19] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, pages 49–63, 2005.

[20] E. Reingold and J. Tilford. Tidier drawings of trees. *IEEE Transactions on Software Engineering*, pages 223–228, 1981.

[21] A. Seary and W. Richards. Spectral methods for analyzing and visualizing networks: An introduction. In *National Resrach Council, Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, 2003.

[22] B. Shneiderman. Tree visualization with tree-maps: 2-d spacefilling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[23] M. B. Sven Gothel. Jogl: Java bindings for opengl. `https://jogl.dev.java.net/`.

[24] S. T. Teoh, K. L. Ma, S. F. Wu, and X. Zhao. Case study: interactive visualization for internet security. In *Proceedings of the conference on Visualization*, pages 505–508, 2002.

[25] W. Torgeson. Multidimensional scaling of similarity. *Psychometrika*, 30:379–393, 1965.

[26] W. Wang and T. E.Daniels. Diffusion and graph spectral methods for network forensic analysis. In *Proceedings of the workshop on New Security Paradigms*, pages 99–106, 2006.

[27] M. O. Ward. Xmdvtool:integrating multiple methods for visualizing multivariate data. In *Proceedings of IEEE Conference on Visualization*, pages 326–333, 1994.

[28] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometric and intelligent Lab. Sys.*, 2:37–52, 1987.

[29] B. Xiao, B. Yu, and C. Gao. Detection and localization of sybil nodes in vanets. In *Workshop on Dependability in wireless ad hoc networks and sensor networks*, pages 1–8, 2006.

[30] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. In *In the Proceedings of the 8th SIAM Conference on Data Mining*, pages 739–750, 2008.

[31] X. Ying and X. Wu. On randomness measures for social networks. In *Proceedings of the SIAM Conference on Data Mining (SDM)*, pages 709–720, 2009.

[32] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *IEEE Symposium on Information Visualization*, 2005.