Sybil Attack Detection through Global Topology Pattern Visualization

Aidong Lu, Weichao Wang, Abhishek Dnyate, Xianlin Hu

Abstract— We present a robust intrusion detection approach for wireless networks based on a new multi-matrix visualization method with a set of pattern generation, evaluation, organization, and interaction functions. Our approach concentrates on assisting users to analyze statistical network topology patterns that could expose significant attack features. Specifically, we investigate Sybil attacks that have severe impacts on the fundamental operations of wireless networks. We have analyzed the features of network topologies under various Sybil attacks and, consequently, designed several matrix reordering algorithms to generate statistical patterns. These topology patterns are automatically evaluated and classified through the measured structural similarities to the signature attack patterns. We have also designed a new time-series analysis method to identify attack durations with a time histogram generation and an automatic segmentation method. To handle complex Sybil attacks, we have integrated our pattern generation, evaluation, and organization methods to construct a prototype detection system, in which specialized interaction functions are provided to assist the analysis and comparison of network data. Simulation results show that this approach can effectively locate Sybil attacks under different combinations of network parameters. Our multi-matrix visualization method provides a flexible framework to handle the intricacies and implications from building a complex visual analytics system, which can be extended to defend against a wide range of attacks.

1 Introduction

With the wide adoption of wireless networks in real-life applications, enforcing security in these environments has become a top priority. Both automated attack mitigation and interactive visualization approaches have been developed for intrusion detection. However, due to the diversity and complexity of malicious attacks, automated attack detection algorithms are often built upon some strong assumptions while interactive visualization methods can overly rely on user intervention, which will restrict their applications to real-life problems. Therefore, the suitable combination of these two types of approaches in order to reveal feature patterns effectively and efficiently is crucial to the development a practical detection mechanism. This paper presents an approach that can automatically suggest interesting data patterns and assist users in identifying malicious nodes with just a small amount of interaction.

In all the intrusion detection mechanisms, several methods have been developed to visualize topologies using graph drawing or matrix representation [2, 17], since topology data is commonly collected in network applications and is extremely important for routing. A global network topology records the neighbor relationships among wireless nodes and includes many traces that can be used to detect attacks on authentication and node identities. Since these traces are usually deeply hidden in the topology information, it is very difficult for a user to identify malicious nodes with existing general topology visualization methods. Therefore, we need to develop more effective interactive visualization and analysis approaches to accomplish these challenging tasks.

Specifically, in this paper we investigate *Sybil attack* [10] that manipulates node identities under various attack scenarios in wireless networks. In such attacks, a single malicious node plays the roles of multiple legitimate members of the network by impersonating their identities or claiming fake IDs. More details about the behaviors and impacts of Sybil attacks are provided in Section 2.1. Since in these attacks the malicious nodes can change the number of fake identities and their connection relationships freely, the effectiveness of previous intrusion detection systems may be drastically weakened [10]. Noticing the serious harm that Sybil attacks can cause, researchers have proposed several approaches to defend against such attacks [5, 8, 10, 24]. Existing approaches usually concentrate on verifying whether or not a pair of nodes have distinct resources, distinct knowledge, or distinct positions. However, these automatic algorithms often make certain assumptions about the environments and are not capable of detecting complex variations of Sybil attacks. We believe that visualization of global topology is one promising direction and it is essential for the development of effective and robust analysis and monitoring approaches that can assist users to detect such attacks.

In this paper, we present a new approach to detect Sybil attacks through exploring statistical patterns of global network topology based on attack features. We have designed a series of pattern generation, evaluation, organization, and interaction methods that allow users to explore and analyze topology data efficiently. Our approach can reveal significant patterns through both matrix reordering and time-series analysis. The designs of multi-matrix visualization and supporting analysis functions allow us to overcome the limits of automatic algorithms with the advantages of interactive visualization techniques. For complex attack scenarios, as long as one of our generated topology patterns shows inconspicuous traces, even if they are fuzzy and incomplete, our interactive exploration process can help users utilize their visual cues, combined with their expertise to identify malicious nodes. This visualization assisted approach provides a practical and robust detection tool by exposing special patterns hidden in the time-varying topology information.

We have summarized several features of Sybil attacks and used them to explore similar patterns hidden in the global topology matrices. New patterns are generated by reordering topology matrices according to different aspects of attack features. We have also designed methods for time histogram generation and automatic segmentation in order to capture attack periods, thus revealing more significant patterns. All the topology patterns are automatically evaluated and categorized to provide detection reports and suggestions to users. To organize and analyze these relevant topology patterns effectively, we provide several specialized analysis tools in our integrated system. Our experimental results show that this method can effectively detect Sybil attacks according to hundreds of simulation datasets.

The main contribution of this paper is to provide a robust intrusion detection and network monitoring approach which focuses on Sybil attacks. Since we integrate attack features with expert knowledge in the detection process, our method can detect complex attack combinations, which are difficult to identify by previous methods. We have designed a series of matrix visualization methods to analyze information using multiple relevant topology patterns based on attack features. We have also provided an integrated method to analyze time-series topology data through designing special time histogram and segmentation methods. Compared to previous security methods, our approach is solely based on neighbor relationships among wireless nodes; therefore it can be applied to highly dynamic environments such as mobile ad hoc networks and can be extended to detect other attacks. Since our system can effectively visualize and organize multiple matrix patterns, we believe that this approach can be extended to serve as integrated analysis and interaction solutions to general network monitoring and attack detection tasks.

The remainder of the paper is organized as follows. We first review previous work on detecting Sybil attacks in networks with both security and visualization assisted approaches in Section 2. Section 3 presents the behaviors of Sybil attacks and our assumptions. Section 4 describes our network pattern generation algorithms based on attack features. Section 5 describes an approach to analyzing time-series topology data with a new time histogram design and an automatic time segmentation method. Section 6 provides our visual evaluation approaches to efficiently categorizing topology patterns generated from the previous two sections. Section 7 presents our integrated visualization system for attack detection and exploration. We present experimental results and discussions in Section 8. Finally, Section 9 concludes the paper and discusses future extensions.

2 Related Work

2.1 Sybil Attack Detection in Networks

A Sybil attack is one particularly harmful attack on distributed systems [6] and wireless networks [10]. This attack has been demonstrated to be detrimental to many important network functions. For example, the Sybil attack is discussed in an architecture for secure resource peering in an Internet-scale computing infrastructure [13]. Newsome et al. [24] have also pointed out that combinations of different types of Sybil attacks may cause severe impacts on wireless sensor networks, which are very difficult to recover.

Existing detection methods can be divided into two categories: identity-based or location-based approaches. The first category generally mitigates Sybil attacks by limiting the generation of valid node information, such as the approach of pre-distributed secret keys [24]. The second category utilizes the fact that each node can only be at one position at any moment, such as the SeRLoc approach that determines node locations passively under known attacks [21].

Since previous approaches rely on the interactions among network nodes within a localized area, they lose the global view, which may be utilized by attackers to present the same fake identities at different places in a network or conduct complex attacks. Our approach studies accumulated global network topologies and detects Sybils by locating anomalies in neighbor relationships and movement patterns of wireless nodes. With the tight integration of interactive visualization and security algorithms, our approach can be used to detect Sybil attacks under more sophisticated scenarios.

2.2 Visualization of Network Topology

With the ever increasing data size and complexity, many visualization approaches have been developed to improve the processing of a large amount of network data including traffic patterns, network flows and logs [4, 23, 30]. Because of the importance of the network topology, it has been used to help enforce Internet and wireless network security in multiple network visualization mechanisms [3, 15, 29]. For example, topologies have been visualized using graph drawing or parallel coordinates [1, 2] to show interesting patterns of malicious attacks.

In this paper, we reorder the time-variant network topology and extract special patterns of Sybil attacks for their detections. We believe that the proposed techniques to model patterns of attacks can be applied to the detection of a broader range of malicious activities.

2.3 Matrix Reordering

While matrix-based representations have been used in a wide range of applications, automatic reordering of matrices is a very challenging problem [9]. Many researchers have proposed to use mathematical or heuristics [27] approaches, such as objective functions [18] and minimum linear arrangement [20], to automatically reorder a matrix. Interactive approaches have also been used to analyze matrix information [14, 25].

Our method in this paper also reorders matrices to reveal important patterns in the network topology. Different from previous approaches, we design automatic reordering and evaluation algorithms according to attack features and use resulting patterns to detect malicious nodes.

3 Challenges of Sybil Attacks

Now, we briefly describe the behaviors of Sybil attacks and their potential harm to a wireless network. As the name of 'Sybil attacks' implies, malicious nodes play the roles of multiple legitimate members in a network by impersonating their identities or claiming fake IDs. These fake nodes do not have real physical devices like legitimate nodes and they often claim to have direct or indirect connections with the malicious nodes that generate them. Here, we borrow the taxonomy defined in [24] and classify the attacks based on the connections among Sybil and legitimate nodes. If the Sybil nodes can directly communicate with other legitimate nodes, it is a direct Sybil attack. By contrast, in an indirect Sybil attack, a malicious device claims to have the paths to reach Sybil nodes so all messages have to go through it. While Sybil attacks seem to be simple, they can affect network performance at different degrees and cause severe harm, such as manipulating the results of localized voting or data aggregation. In the worst case, Sybil attacks can enable malicious nodes to take over the control of the whole network [10] and defeat the replication mechanisms in distributed systems.

The main difficulties in detecting Sybil attacks come from various combinations of individual attacks. While it is difficult to link together multiple fake identities that appear in different periods of a network's lifetime and detect non-simultaneous attacks, their impacts on network security are also limited. For example, a Sybil node that is not a member of a network cannot cast a vote during the leader election procedure. Therefore, in this paper, we focus on the simultaneous Sybil attacks. To evaluate our proposed mechanism in a more realistic environment, we assume that both direct and indirect attacks exist in the network and a malicious node can dynamically switch between the two types. We also assume that multiple malicious physical devices co-exist in the network and a Sybil node can switch among them.

The challenge also increases exponentially with the number of malicious nodes. Since malicious nodes may freely change the number of Sybil nodes and their connection relationships, the effectiveness of previous intrusion detection systems can be drastically weakened. We believe that an interactive visualization method under the user guidance is necessary for detecting various complicated attack scenarios.

4 Network Topology Pattern Generation

Our detection approach is achieved through a series of pattern generation, evaluation, organization, and interaction methods. This section presents our pattern generation methods based on attack features. We first describe how we collect network topology information from wireless nodes. Then, we present the topology patterns that can be used as indications of Sybil attack existence. These patterns serve as the "signature" of Sybil attacks and are used to guide our interactive analysis process. According to these patterns, we design several automatic generation algorithms from each attack feature. The results of this section provide effective visual information to further assist users in the detection process.

4.1 Global Network Topology Patterns

Since Sybil attacks do not demonstrate anomalies in neighbor relationships at individual time steps, we need to collect the connectivity information among wireless nodes for a time period to detect Sybil nodes. Assume there are N nodes in the network and the time range is $[0,R]$. At each sampled time step, the connectivity relationship can be represented as an $N \times N$ topology matrix T, with $T(i,j)$=1 indicating the connection between node i and node j. In this way, the information of network topology across a time period can be represented as a 3D $N \times N \times R$ connectivity matrix. We can use central controllers that are

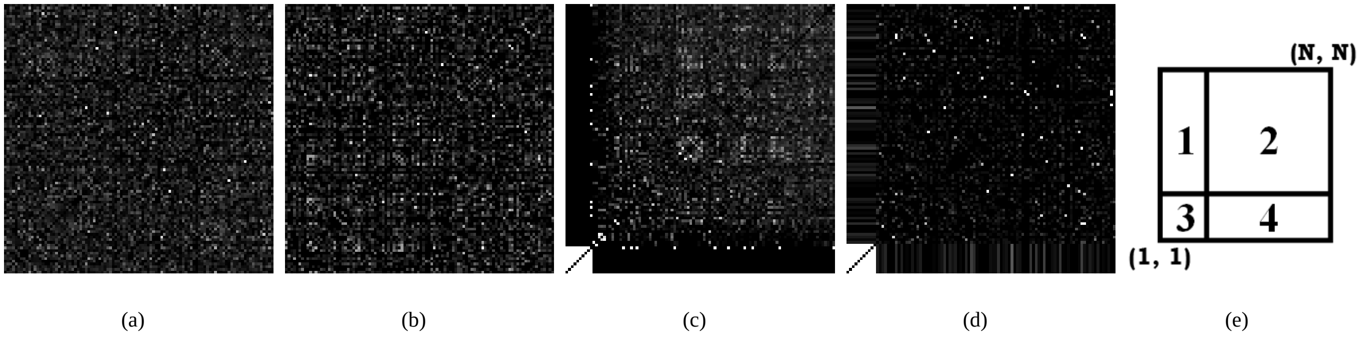


Fig. 1. (a,b) General 2D statistical topology matrices do not reveal any suspicious patterns; (c) Signature pattern for indirect Sybil attacks; (d) Signature pattern for direct Sybil attacks; (e) A 2 by 2 grid structure in the patterns, with index (1, 1) at the left bottom corner.

special nodes in a wireless network to collect network topology information. We summarize the 3D connectivity matrix into a 2D global topology table, which records the number of time steps that each pair of nodes are connected during the time period under study. We choose to concentrate on analyzing 2D global topology patterns, since they are convenient for users to visualize.

The signature patterns of Sybil attacks are found to be 2 by 2 grid structures, as shown in Figure 1. Generally, a topology pattern across any time period appears to be random without special arrangements (Figure 1 (a) and (b)). When we reorder the node sequence, we may see some interesting 2 by 2 grid structure patterns, as shown in Figure 1 (c) and (d). These two special patterns are closely related to the attack procedures and indicate the existence of malicious nodes. Simply speaking, Sybil attacks can be summarized as a malicious device presenting multiple identities to the network. There are two types of Sybil attacks: direct attacks, in which malicious nodes use multiple fake identities to directly communicate with other nodes; and indirect attacks, in which a malicious device claims to have the paths to reach the Sybil nodes and all messages have to go through it. Because of the time and location constraints, similar signature patterns are exposed when malicious nodes and legitimate nodes are separated in the 2D statistical matrix. According to the pattern features, we have developed several automatic arrangement methods to expose patterns that are similar to the signature patterns. These new patterns will be used to detect attacks later. To illustrate our pattern generation algorithms, we divide topology patterns into 2 by 2 grid structures, as shown in Figure 1 (e).

4.2 Pattern Generation

We design automatic algorithms to expose the patterns hidden in the global topology matrix that are similar to the signature patterns of Sybil attacks. Our approach is to generate new patterns by reordering node sequences along the two dimensions of the global topology matrix. Since the 2D topology matrix of a network with N nodes may generate $N! \times N!$ different patterns, it is obviously too time consuming for users, such as network administrators, to manually adjust node sequences. Therefore, we need to automatically arrange node sequences during the decision making process, especially for complex attack scenarios and large scale networks.

We use the features of Sybil attacks to guide our automatic pattern generation processes. We have analyzed these attacks from multiple aspects and designed matrix reordering algorithms according to each attack feature. These patterns are then automatically evaluated and organized for users to detect attacks interactively. Generally, we can declare the existence of attacks as long as one of the reordered sequences shows a suspicious pattern. This approach allows us to analyze the 2D global topology matrix from multiple independent or correlated aspects. We show in our results that this method is convenient and robust for detecting various Sybil attack combinations. The following describes four automatic pattern generation methods that have been found to be effective in detecting our signature patterns.

4.2.1 Method 1 - Anchor Connection

Our first method is designed for indirect attacks according to the connection feature of Sybil nodes and legitimate nodes. As shown in the signature pattern of indirect attacks in Figure 1 (c), there are two blocks that are almost empty: regions 1 and 4. This statistical feature is a result of the lack of direct connectivity between Sybil and legitimate nodes and long-time connections among the Sybil nodes. Corresponding to the attack definition, indirect Sybil nodes can communicate with legitimate nodes only through a small number of malicious 'anchor' nodes. Although this may not be obvious at a single time step, it becomes more and more visible in the statistical matrix with the increasing length of the monitored time duration. Our first method is designed to reorder the global topology matrix to form such patterns through the following procedure:

1. For each row, measure its connectivity degree by accumulating the square of data values;
2. Sort rows in decreasing order of connectivity degrees;
3. Apply the row sequence to the column.

Since the initial global topology matrix is symmetric, we can apply the row sequence to the column directly to sort the connectivity degrees. As shown in the row titled 'Method 1' in Figure 2, this method successfully captures this feature of indirect attacks.

4.2.2 Method 2 - High Connectivity

Our second method is designed for both types of Sybil attacks according to the high connectivity feature among fake identities. As shown in Figure 1 (c) and (d), the left bottom corner (region 3) of our signature patterns accumulates a block of bright pixels. This indicates the existence of a group of highly connected nodes in the network. Corresponding to the nature of these attacks, since multiple Sybil nodes are fabricated by the same physical device, their locations are usually close to each other. These malicious nodes often have to claim that they are connected to avoid being detected by location-based methods. We design this method to form a large value block at the left bottom corner in a global topology matrix through the following procedure:

1. Repeat the steps 2-4 from $m = N$ to $m = 2$ to reorder the whole pattern.
2. Scan the top right $m \times m$ region and select one item (i, j) with the largest value;
3. Switch row $N - m + 1$ and row j;
4. Switch column $N - m + 1$ and column i;

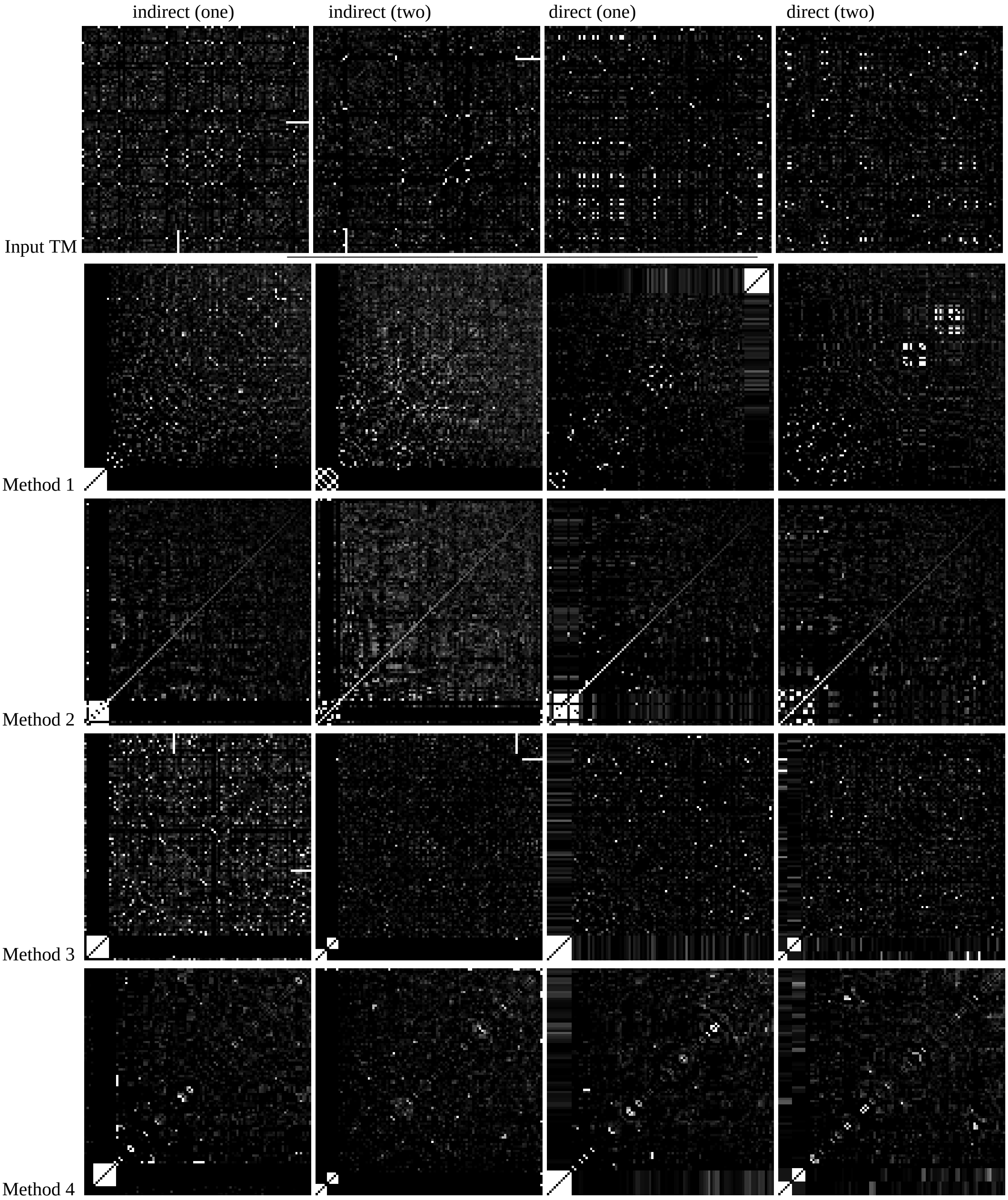


Fig. 2. Examples of our pattern generation results. The first row shows four 2D statistical topology matrices and the second to fifth rows show their corresponding reordered patterns from methods 1 to 4. These four datasets contain one group of indirect attack nodes, two groups of indirect attack nodes, one group of direct attack nodes, and two groups of direct attack nodes respectively. Most reordered topology matrices demonstrate more obvious attack patterns than their initial matrices.

This feature is especially useful in detecting direct attacks since the fake identities communicate directly with legitimate nodes, who will report all the connections honestly. Even if attackers increase the number of Sybil nodes to reduce their average connection number, they still need to keep high adjacency values for the attack effectiveness. The row titled ‘Method 2’ in Figure 2 shows that this method is useful for the detection of both direct and indirect attacks. Although these patterns may not be as obvious as our signature patterns, they are clear enough for users to capture this feature.

4.2.3 Method 3 - Close Locations

Our third method is designed for both types of Sybil attacks according to the moving feature of Sybil nodes. As shown in the signature pattern of direct attacks in Figure 1 (d), regions 1 and 4 demonstrate clear horizontal and vertical band patterns. Actually, the empty regions 1 and 4 of indirect attacks in Figure 1 (c) can also be viewed as a special case of these band patterns. Corresponding to the attacks, this feature indicates similar movement patterns of Sybil node groups, while legitimate nodes rarely share the same moving trace for a long time duration. This is inevitable due to the fact that Sybil nodes are attached to the same physical device. At a single time step, we can input the topology matrix to a multi-dimensional scaling (MDS) method [28] to reconstruct the physical distances among the wireless nodes in a network. Similarly, the reconstructed locations from a statistical global topology matrix can be used to measure average node distances in a time duration. With the similar moving patterns of Sybil nodes, we can use the distribution of reconstructed locations to separate malicious nodes from legitimate ones. Figure 3 shows two examples of malicious nodes separated from the legitimate nodes. We design this method to group the nodes based on their reconstructed locations from the statistical matrix:

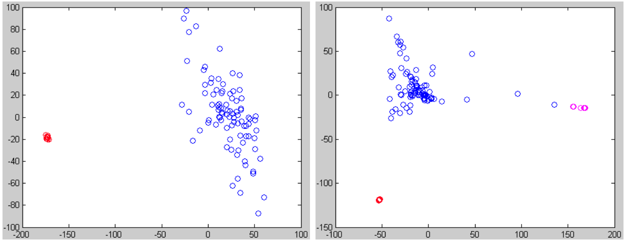


Fig. 3. MDS reconstructed node locations can be used to detect Sybil attacks, since malicious nodes tend to move in groups during a time period. Legitimate nodes are colored blue and two malicious groups are colored red and purple, respectively.

1. Calculate a dis-connectivity matrix by reversing the global topology matrix: $D(i,j) = 1 - T(i,j)$;
2. Reconstruct 2D statistical node locations using MDS method;
3. Calculate the center position of all the nodes;
4. Reorder the sequence of nodes according to their distances to the center position in a decreasing order;
5. Apply the sequence to both row and column.

The row titled ‘Method 3’ in Figure 2 shows that this method is useful for both direct and indirect attacks. We expect that clustering algorithms can be applied to improve this algorithm.

4.2.4 Method 4 - Similarity of Neighbors

The fourth method is designed for both types of Sybil attacks according to the neighbor similarities of a malicious node group. This feature is also represented by the band patterns in regions 1 and 4 in Figure 1 (c) and (d). Since only the nodes within a limited range can hear each other in a wireless network, malicious nodes often share common neighbors for a long duration. This method is especially effective in detecting direct attacks, since fake identities can directly communicate to legitimate nodes. In indirect attacks, although malicious nodes can manipulate their neighbor lists by removing some fake identities, their choices are restricted by the total number of neighbors. We design the following procedure to reorder a global topology matrix according to the information of neighbor similarities.

1. Calculate the similarity matrix $S(i,j)$ by summarizing the number of common neighbors of every node pair in a time duration: $S(i,j) = \sum_t \sum_{k=1}^{N} (T(i,k) \cdot T(j,k))$
2. Select an item $S(i,j)$ with the largest similarity value;
3. If neither row i nor j has been reordered, place the rows i and j after previously placed nodes starting from row 1;
4. Otherwise if row i or row j has been reordered, insert the other row right after the previously reordered row;
5. Repeat steps 2-4 until all rows have been reordered.
6. Apply the same sequence to the column.

As shown in the row titled ‘Method 4’ in Figure 2, this method generates obvious patterns for both Sybil attack types.

5 Time-Series Analysis

It is often critical to analyze the changes of a network over time for intrusion detection. We are interested in detecting the durations of attacks, especially the starting and ending time steps, that play important roles in revealing hidden topology patterns. Our approach is to design a time histogram, which can guide users in analyzing time sections through exposing certain data features. For this new time histogram, we also provide an interaction function for users to select attack durations, as well as an automatic segmentation algorithm based on attack features. This integrated approach has been shown to be robust in detecting complex Sybil attacks.

The time-series analysis is especially important for attacks that cannot be detected based on the information of a single time step. For such attacks, we need to analyze the network information from a certain time duration. Due to the diversity and complexity of Sybil attack combinations, it is very challenging to identify their accurate starting and ending times. Simply using all the collected time steps or dividing time sections equally will not work well. If a time section is much longer than the contained attack duration, the generated patterns may be masked by normal activities of legitimate nodes; otherwise if a time section is too short, we may not have enough information to generate useful patterns. Therefore, we need a robust approach to select appropriate time ranges for revealing attack patterns.

The motivation of using time histogram is to provide users a mechanism to find suitable time segmentation by revealing attack features in the 2D node-time space. As shown in the widely adopted histogram (data value and gradient magnitude) for volume visualization [19], the histogram is a very powerful tool to convert data from any dimension to a 2D space, in which it is convenient for observation and interaction. In security visualization, the closest research is the IDGraphs by Ren et al. [26] that visualizes the space composed by the number of unsuccessful connections vs. time. All these histograms are much more intuitive for users to study than those raw datasets. Different from previous approaches, we generate the time histogram with a specially designed algorithm based on attack features. To the best of our knowledge, no other work has addressed histograms to this degree.

5.1 Time Histogram Design and Interaction

We design a 2D histogram to reflect data properties along the time axis based on attack features. This time histogram can also assist users to analyze attack durations. Since fake identities in both types of Sybil attacks usually appear and move in groups during a certain time period, they share a large portion of neighbors for multiple time steps. Our histogram collects this grouping information and produces obvious patterns along the time axis.

Our time histogram is a grey scale image in the space of node index and time step. For every node at each time step, we measure its “significance” value corresponding to the attack features, and generate a histogram by linearly mapping all the significance values onto grey colors. The significance values are calculated by the following procedure. At each time step, we group nodes together if they share the same set of neighbors. With this method, each node belongs to only one group at each time step. The significance value of a node is set as the size of its group divided by the largest group size at this time step. We paint larger significance values with brighter colors. The time histogram is generated by traversing all the time steps.

Figure 4 shows an example of our time histogram. Since Sybil nodes move in groups for a duration that is long enough to harm the network, they form obvious line strips in the histogram. Even when these malicious nodes try to hide themselves through appearing on and off, patterns similar to isolated line strips can still be detected by human eyes. On the contrary, normal nodes move randomly and will not generate the stripe patterns. When several normal nodes do move in groups, their patterns in the histogram cannot be distinguished from malicious nodes. We need to introduce expert knowledge through human interaction to detect this and other complex combinations of Sybil attacks.

We also implement a simple drawing function that allows users to specify time transfer functions. As shown in Figure 4, normal time steps are covered with a blue mask and suspicious time steps are covered with a brown mask. Users can drag the mouse to specify the normal/suspicious time durations and their starting and ending time steps repeatedly. After selection, users can choose to generate topology patterns for all the suspicious time periods for further analysis.

5.2 Automatic Segmentation Algorithm

We also design an automatic segmentation method to detect the existence of Sybil nodes. This method is not to replace the role of human experts; instead, it is to suggest suspicious activities, reduce interaction burdens, and re-examine detection results.

Ideally, we want to separate the neighboring time steps when one of the three events happens: start of an attack, end of an attack, or changes of attacks. These three events cannot be detected simply based on the similarities between adjacent time steps. We design a method that uses the statistical results from the entire time period and the grouping features of Sybil attacks. Specifically, during the process of an attack, if adjacent time steps contain similar suspicious node groups, we merge them. When there are no suspicious node groups, we always merge these time steps. When suspicious node groups change, such as the addition of new groups or the disappearance of existing groups, we separate the time steps. The following describes a score table that captures the degree of suspiciousness of wireless nodes and our segmentation method based on the grouping feature.

We first calculate a score table for each node pair $TS[i][j]$ to measure their suspicious degrees using information from the entire monitored duration. To filter out small noises (when two nodes happen to move together), we choose a small time window ranging from 20 to 100 time steps to divide monitored duration equally. The topology matrix of a time window is the average of all the contained topology matrices. Since this reduces the number of time steps for the segmentation process, it also significantly improves the computation efficiency. For each time window starting from the first, we modify the score table by the following equation:

$$TS[i][j] = \prod_{k=1}^{n} (1 + weight * groupsize_k(i, j)) \qquad (1)$$

Here $groupsize_k(i, j) = 0$ if i, j are not in the same group in window k; otherwise, it is the number of nodes in the group. In this paper, we use $weight = 0.001$. In this way, the score of suspicious node pairs increases exponentially to stand out from legitimate nodes. After we traverse all the time windows, we normalize the score values to the range of [0, 1] for further analysis.

Second, we segment the entire time duration by deciding whether or not we should merge or separate adjacent time windows. Starting from the first two time windows, for every adjacent pair, we calculate the intersection of their node groups. We also measure the accumulated score for a time window t as

$$GroupsScore(t) = \sum_{g_k} \sum_{i,j \in g_k} TS[i][j] \qquad (2)$$

Here g_k are node groups in window t, and i and j are nodes in g_k. Assuming TA and TB are two adjacent time windows, we adopt the following method to determine whether or not we should merge or separate them. According to the group scenarios, there are three cases:

1. When neither TA nor TB has groups, we merge TA and TB.
2. When only one window has groups, we merge TA and TB if the GroupsScore of this window is smaller than $thres1$, which is a user-assigned parameter; otherwise we separate them.
3. When both TA and TB have such groups,
   (a) If both GroupsScores are smaller than $thres1$, merge;
   (b) Else if only one GroupsScore is smaller than $thres1$, separate them;
   (c) Else we calculate the intersection of the groups of TA and TB. Then, if $(GroupsScore(TA \bigcap TB)$ / $GroupsScore(TA) > thres2)$ and $(GroupsScore(TA \bigcap TB)$ / $GroupsScore(TB) > thres2)$, merge; otherwise separate them.

We use 1.0 and 0.5 as the threshold values $thres1$ and $thres2$ respectively. As shown in Figure 4, the transfer function is generated with our automatic segmentation result, which captures the attack duration. Since the segmentation is calculated based upon time sections, it may not reflect the accurate starting and ending time steps of the attack. We can decrease the length of time sections for more accurate results at the expense of heavier computation overhead. The histogram is generated within a minute and the segmentation process takes 5 to 8 minutes for data with 10000 time steps. This method has been shown to be robust enough to reveal obvious topology patterns in our later experiments.

6 Automatic Pattern Evaluation and Organization

To handle multiple topology patterns efficiently, we have developed automatic pattern evaluation and organization methods. All the topology patterns generated from the previous two sections are automatically evaluated and organized in our system, so that users can efficiently analyze and acquire useful information. This can also significantly accelerate the detection process by reducing unnecessary user interaction. We describe the evaluation and organization methods in this section and our integrated detection system in the next section.

Since topology patterns in real life may appear in various formats, the most effective way to evaluate topology patterns is to assess their structure similarities to the signature attack patterns. We need to compare the structures of patterns to the features of two signature patterns, since both data values and region sizes may vary under attacks. We have designed automatic evaluation methods for both types of Sybil attacks that return scores between 0 and 1 for each pattern. Since the signature patterns indicate existence of attacks, the patterns with large scores have a high probability to be under attack.

To assess the structure similarity between an arbitrary pattern and a signature pattern, we build our automatic evaluation methods with two steps. The first step tries to construct a 2 by 2 grid structure as our signature patterns and the second step measures the consistency of data distributions according to signature pattern features. The following describes the pattern features and our evaluation algorithms for direct and indirect attacks respectively.

6.1 Pattern Evaluation for Indirect Sybil Attacks

As shown in Figure 1 (c), the signature pattern of an indirect attack possesses a 2 by 2 grid structure, with large values in region 3, small values in region 2, and two almost empty areas in regions 1 and 4. Our pattern matching procedure is designed as follows:

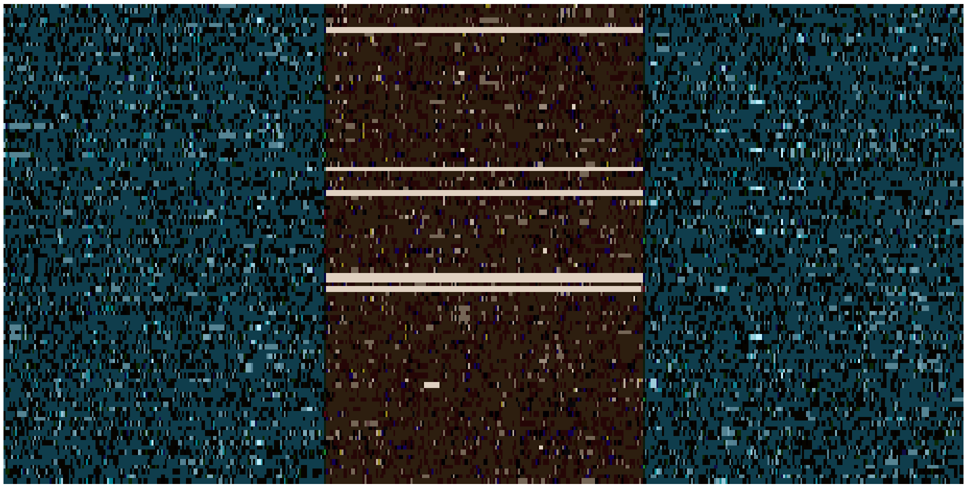


Fig. 4. An example of time histogram. The background is a grey scale time histogram with brighter colors suggesting more suspicious nodes and time steps. The line strips in the middle indicate possible attacks and can be used to identify attack durations. The blue/brown masks are time transfer functions generated by our automatic segmentation method. Users can also design new or modify existing time transfer functions with the system interface.

1. Detect a 2 by 2 grid structure in the pattern. Assume our dividing lines are row r and column c, which divide an $N \times N$ image into 4 regions: region 1 ($c \times (N-r)$), region 2 ($(N-c) \times (N-r)$), region 3 ($c \times r$), and region 4 ($(N-c) \times r$). We start the dividing lines from the left bottom corner with row 1 and column 1 and calculate the average values of regions 1, 3, and 4 respectively. The dividing line is moved from column c to $(c+1)$ if this change leads to a higher ratio of $\frac{average3}{average4}$. Similarly, the dividing line is moved from row r to $(r+1)$ if this leads to a higher ratio of $\frac{average3}{average1}$. Finally, the dividing lines are located at the two locations with the largest area ratios.
2. Calculate the average values of regions 2 and 3 as average2 and average3 respectively;
3. $Score1_{id}$ = (1 + average3 - average2) / 2;
4. $Score2_{id}$ = (number of zeros in region1 and region4) / (area of region1 and region4);
5. $Score_{id} = w_1 * score1_{id} + w_2 * score2_{id}$.

In our implementation, we use 0.3 for w_1 and 0.7 for w_2 since features of regions 1 and 4 are more difficult to hide by malicious nodes.

6.2 Pattern Evaluation for Direct Sybil Attacks

Similar to the signature pattern of indirect attacks, the signature pattern of direct attacks can also be divided into a 2 by 2 grid structure, with large values in region 3 and small values in region 2. The major difference is that region 1 demonstrates similar column patterns and region 4 demonstrates similar row patterns, as shown in Figure 1 (d). This results from the close positions of Sybil nodes attaching to the same physical device. We modify our evaluation method for indirect attacks to accommodate the signature pattern differences.

1. Detect a 2 by 2 grid structure in the pattern. Assume our dividing lines are row r and column c, which divide a pattern into 4 regions as for indirect attacks. We start the dividing lines from the left bottom corner with row 1 and column 1. We calculate the average values of regions 1, 3, 4 and adjacent column similarity in region 1, and adjacent row similarity in region 4. The dividing line is moved from column c to $(c+1)$ if this change leads to a higher ratio of $\frac{average3}{average4}$ or preserves a high similarity value of the new region 4. Similarly, the dividing line is moved from row r to $(r+1)$ if this leads to a higher ratio of $\frac{average3}{average1}$ or preserves a high similarity value of the new region 1. We repeat this process for all the rows and columns;
2. Calculate the average values of regions 2 and 3 as average2 and average3 respectively;
3. $Score1_d$ = (1 + average3 - average2) / 2;
4. $Score2_d$ = (adjacent column similarity of region 1 + adjacent row similarity of region 4) / 2;
5. $Score_d = w_1 * score1_d + w_2 * score2_d$.

We use 0.2 for w_1 and 0.8 for w_2 to emphasize the similarities in regions 1 and 4.

6.3 Automatic Pattern Organization

To automatically organize all the topology patterns, we use both evaluation methods to assess all the four patterns of a time period, resulting in eight scores. Since each pattern generation method represents one attack feature, as long as one of the eight scores is larger than a user-defined threshold, we report the existence of attacks in the network.

Table 1 shows our evaluation results for the sample patterns in Figure 2. The results demonstrate that multiple patterns of the same dataset may receive high scores, which indicates that Sybil attacks can be successfully detected from multiple aspects.

We further use the evaluation scores to classify each pattern into one of the following categories: indirect attacks, direct attacks, and uncertain/safe. Since empty regions in indirect attack patterns can be viewed as special cases of similar rows or columns, the datasets under indirect attacks also receive high scores from the method for direct attacks. When we try to distinguish between direct and indirect attacks, a pattern that receives a high score from the evaluation method for direct attacks and a low score from the method for indirect attacks will be labeled with high risks of direct attacks; while a pattern that receives high scores from both methods will be labeled with high risks of indirect attacks. This classification provides some analysis results to help users make final decisions.

7 Integrated Detection System

We integrate all the methods from the previous sections with several interaction schemes to construct a detection system that can assist users in exploring unknown Sybil attacks and adjust detection strategies accordingly. The system also serves as a monitoring and summarization tool that can report the existence of attacks and provide detailed information of suspicious nodes.

Table 1. Evaluation results of sample patterns in Figure 2. Large scores represent high risks under Sybil attacks.

Scores for indirect attacks	indirect 1	indirect 2	direct 1	direct 2
method 1	0.972745	0.924494	0.0	0.0
method 2	0.923721	0.808646	0.432958	0.584031
method 3	0.898569	0.931796	0.464852	0.495726
method 4	0.915566	0.930933	0.543349	0.596565

Scores for direct attacks	indirect 1	indirect 2	direct 1	direct 2
method 1	0.97185	0.843884	0.0	0.0
method 2	0.835537	0.635931	0.767646	0.855556
method 3	0.834491	0.854223	0.987547	0.820318
method 4	0.854756	0.853441	0.987547	0.965267

	indirect 1	indirect 2	direct 1	direct 2
Maximum	0.972745	0.931796	0.987547	0.965267

7.1 System Design

We have developed an integrated system to detect Sybil attacks through visualizing and analyzing multiple reordered topology patterns. Since visual patterns are often easy to understand, they provide a powerful interaction domain for monitoring and detection tasks. As shown in Figure 5, our interface is composed of two regions: the top pattern window and the bottom interaction and information window. The pattern window is used to visualize topology patterns under three categories (direct attack, indirect attack, and uncertain/no attacks). Users can specify their desired pattern order for each category with provided sorting methods. The default order is set as the decreasing order of evaluation scores to direct users' attention to the most distinguishable patterns. Since the window space can display only several images for each category, we provide browsing buttons on each side. Users can double click an image in the pattern browsing window, and the system will highlight it with red boundaries and enlarge it for better observation. Users can also identify or suggest legitimate and malicious nodes in the enlarged pattern. The middle suggestion panel provides a list of suspicious nodes and their suspicious degrees calculated from relevant topology patterns. The right interaction panel groups contain important interaction functions to regenerate topology patterns, reorder images, divide time sessions, handle data directories, etc.

7.2 Interaction Tools

We have developed several interaction tools to assist users to detect Sybil attacks through analyzing topology patterns. Here we concentrate on the interaction tools that are important to the detection procedures.

First, our system provides a list of suspicious nodes. If there are patterns having evaluation scores larger than the user-defined thresholds, our system will automatically extract a list of suspicious nodes from these topology patterns. We collect information from all the patterns under the two attack categories (these patterns have attack evaluation scores larger than the corresponding evaluation thresholds). For each pattern, we use the corresponding evaluation method of the pattern category to identify the malicious nodes on the left bottom corner. Since the patterns may be asymmetric, we collect information along both dimensions. The number of patterns that detect the same suspicious node is also recorded to suggest its suspicious degree. Both the list of suspicious nodes and their suspicious degrees are shown in the suggestion panel of our system.

Second, we provide an interaction tool to allow users to incorporate their knowledge and assumptions. Users can identify or suggest legitimate and malicious nodes in the enlarged image window. This selection can be made on both dimensions and the selection results are highlighted as red lines for malicious nodes and green lines for legitimate nodes. We incorporate user selections by moving malicious nodes to the left bottom corner and legitimate nodes to the right top corner. We regenerate all the topology patterns by applying pattern generation methods to the rest of the nodes and update the system with the new patterns. This interaction tool is especially useful to assist users to locate all the malicious nodes through only a few rounds of selections.

Third, our system allows users to analyze multiple topology patterns with sorting and moving tools. We provide several automatic sorting tools to change the pattern order. We can sort patterns according to the increasing or decreasing order of their evaluation scores. If a user selects a particular pattern, we can sort patterns based on their similarities to the selected one. If a user selects a malicious node, we can sort patterns by listing those patterns that detect this malicious node first. These sorting and moving interactions help users group relevant patterns together and identify suspicious nodes.

Fourth, we have developed several analysis tools to assist with the interactive detection process. Our system allows users to adjust the alarm thresholds of evaluation scores to reorganize topology patterns under the three categories and regenerate the list of suspicious nodes. We also allow users to select their interested time ranges by specifying the starting and ending time steps. The parameters to form time sessions can also be adjusted to change the time segmentation algorithm. These settings are important to the detection process and users can use these tools to adjust the detection algorithm.

8 Experimental Results

8.1 Simulation Setup

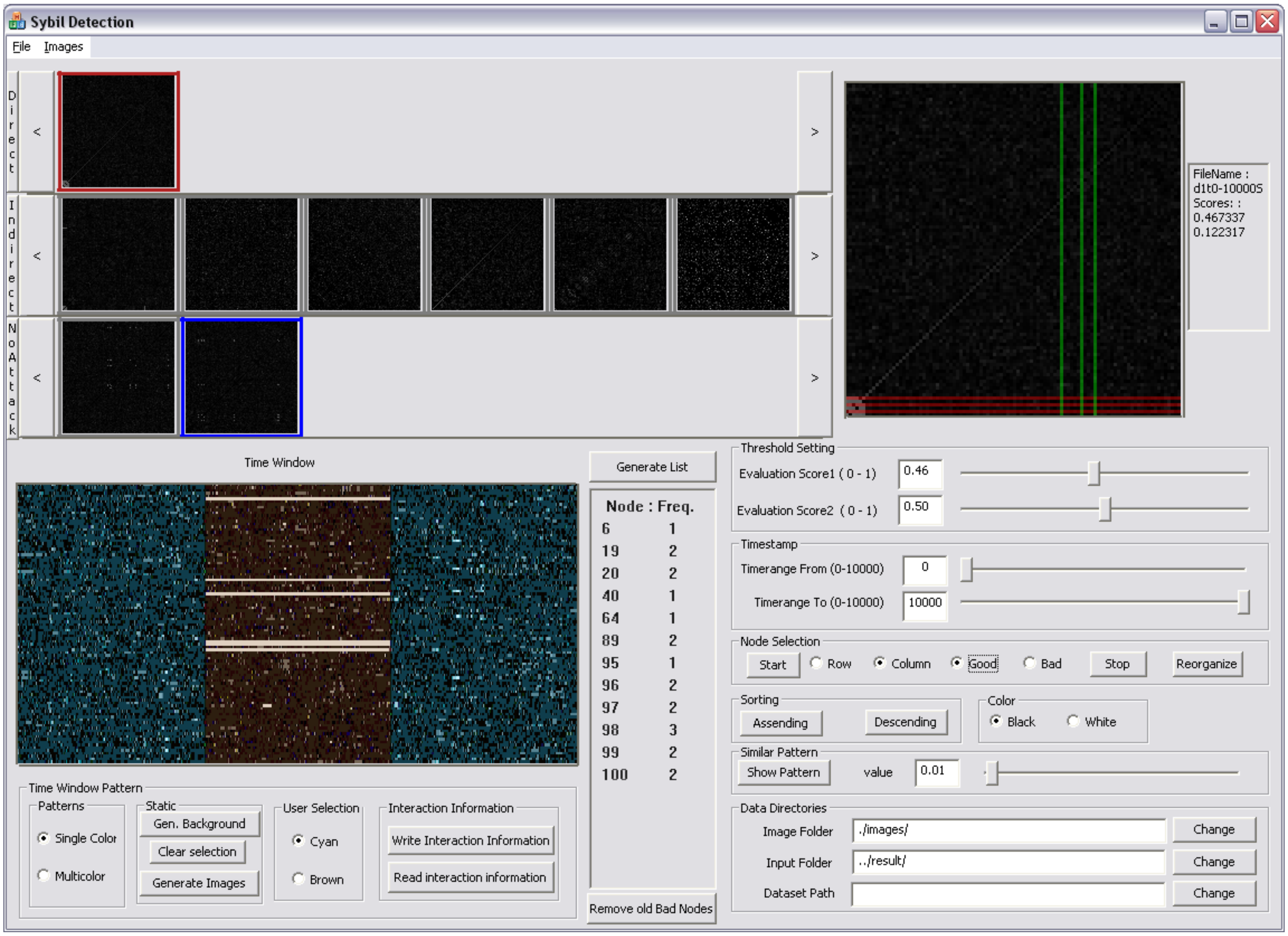
We use simulation to evaluate the proposed mechanism and its capabilities to process and analyze time-dependent topology information collected from a wireless network. We assume that one hundred nodes (including Sybils) are deployed in a $1400m \times 1400m$ area. We adopt the weighted random waypoint model [16] to generate the independent movement patterns of wireless nodes. We assume that a special node exists in the network, which is called the "controller". It can integrate, process, and visualize network topology data and analyze information that is collected from the wireless nodes and the Sybils. We assume that the controller has the storage and computation resources that are needed for the proposed mechanism. In our studies, we use a PC with 3.0GHz CPU as the controller, which can process the information of a network containing several hundred nodes in real time. We assume that each wireless node has established a pair-wise key with the controller. This task can be accomplished during the network initiation procedure or based on some pre-distributed information [7, 11, 12, 22].

We assume that there are multiple malicious physical devices in the network. A malicious physical device can generate up to five Sybil nodes through direct or indirect attacks. A Sybil node can dynamically switch among malicious physical devices during the network lifetime. To establish a comparison standard, we also apply our mechanism to a wireless network that does not contain any Sybil nodes.

We assume that two nodes are neighbors when the distance between them is shorter than r, where r is defined as the radio range. Connection links among wireless nodes are bidirectional. Two communication ranges with the values of $200m$ and $300m$ are adopted in our simulation. We experiment with different highest moving speeds ranging from 5 m/s to 20 m/s, which cover the speed from human jogging to vehicle riding in country field. Different combinations of radio ranges and highest moving speeds are investigated through simulation.

8.2 Simulation Results

We experiment with 24 different combinations of network parameters: 2 radio ranges, 3 highest moving speeds, and 4 switching frequencies of Sybil nodes. For each combination, multiple attack scenarios are generated, including no attacks, one or two groups of direct Sybil attacks, and one or two groups of indirect Sybil attacks. For each case, five independent moving patterns are generated. In total, we evaluate our detection method with 510 datasets. Figure 6 plots the false positive and false negative rates of our detection method in the simulations when the evaluation threshold changes from 0.1 to 1.0. It is clear that when the threshold is 1.0 no attack can be detected, thus the false positive rate is 0 and the false negative rate is 100%. Similarly,



FileName : d1t0-100005
Scores: : 0.467337 0.122317

Node	Freq.
6	1
19	2
20	2
40	1
64	1
89	2
95	1
96	2
97	2
98	3
99	2
100	2

Evaluation Score1 (0 - 1) 0.46
Evaluation Score2 (0 - 1) 0.50
Timerange From (0-10000) 0
Timerange To (0-10000) 10000
Node Selection: ○ Row ● Column ● Good ○ Bad
Color: ● Black ○ White
Patterns: ● Single Color ○ Multicolor
User Selection: ● Cyan ○ Brown
Similar Pattern value 0.01
Image Folder ./images/
Input Folder ../result/
Dataset Path

Fig. 5. Our system interface. The top left portion visualizes multiple topology patterns, which are generated using our four pattern reordering methods for each time period. They are organized automatically into three categories (direct attack, indirect attack and no attack) with two pattern evaluation methods. Users can browse, change pattern sequences, and switch pattern categories to group several related patterns together for analysis. The top right is an enlarged image window with detailed information on the right for observing and identifying network nodes. This panel allows users to identify malicious/legitimate nodes as red/green lines. The bottom portion contains our time analysis window and the bottom right portion, a suggestion window showing a list of suspicious nodes and their suspicious degrees, and a group of interaction panels to assist the two panels on the top for generating and analyzing topology patterns.

when the threshold is very low, the false negative is 0 and the false positive is close to 100%. When the threshold is selected appropriately, both false alarm rates can be low. As shown in Figure 6, our method achieves very low false negative and false positive rates when the evaluation threshold is between 0.7 and 0.92. This shows that our proposed detection method has a high detection accuracy.

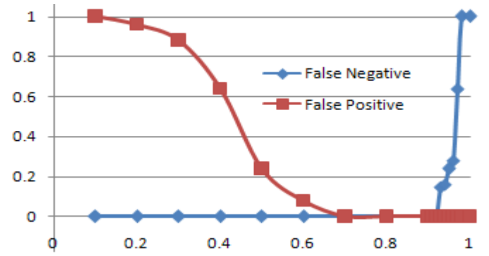


Fig. 6. False positive and false negative curves. Our method achieves very low false negative and false positive rates when the evaluation threshold is between 0.7 and 0.92.

8.3 Case Studies

For simple attack scenarios, such as the example shown in Figure 5, our method can detect Sybil attacks fast and accurately. With the provided time histogram, users can quickly locate the attack period and specify it by drawing on the time window. Our system then runs the pattern generation methods for the time period, calculates evaluation scores, and arranges topology patterns with high suspicious degrees under the categories of attacks. Users can search for signature patterns visually, especially from the patterns which appear first under the attack categories. They can study a particular pattern in the enlarged window, select innocent or suspicious nodes by clicking on the image, and reorder topology patterns with this information. As shown in Figure 5, the time histogram contains obvious strip patterns in the middle, which indicate the attack duration. The pattern under the first category is also similar to the signature pattern for indirect Sybil attacks, and we can label all the nodes on the left bottom corner as suspicious. According to these two clues, malicious nodes can be detected immediately.

We also evaluate the interactive detection process for cases that are difficult to solve by automatic detection algorithms. More user interactions are often involved to provide expert opinions and detect variations of attacks. For example, users may manually adjust suspicious time durations and select suspicious or legitimate nodes according to the traces in the topology patterns. A few rounds of interaction may be needed before patterns that are similar to the signature patterns are found. During this process, our pattern generation algorithms incorporate inputs from users to reorder topology patterns automatically. Therefore, users only need to compare generated patterns with signature patterns, and make hypothesis or conclusions based on the clues in the topology patterns and time histogram. It is worth mentioning that as long as one suspicious pattern is found, our approach can locate all the related attacks immediately by revealing similar signature patterns. This result elicits the detection of attacks or simplification of the attack scenarios. For example, the attacks on the fourth column in Figure 2 are more difficult to detect than those on the first column, since they are less similar to the signature patterns. We can still see some bright blockish regions which may be reordered to form signature patterns. We then label nodes from these regions as suspicious and regenerate all the patterns. A few rounds of labeling and pattern regeneration lead us to locate the first attack group. We may remove them or put them on the left bottom corner, thereby simplifying the attack scenario to only one attack group, which can be detected by our automatic algorithms immediately. This example demonstrates that our interactive detection process is more capable of handling complex Sybil attacks than automated approaches.

8.4 Discussions

The design of multi-matrix visualizations plays the major role in enabling the capability of detecting complex Sybil attacks. Since malicious nodes can always alter their attack strategies according to the detection algorithms, it is extremely difficult to generate an automatic algorithm with fixed, uniform detection procedures. We have considered two aspects to address this problem. One is to detect hidden patterns in the essential global network topologies; and the other is to utilize the exploration and analysis functions of interactive visualization. Matrix visualization allows users to detect subtle patterns using visual cues and user expertise. Our pattern evaluation and organization methods further enhance its capability to reveal hidden patterns from different aspects of pattern features. This integrated approach overcomes the limits of algorithmic methods and provides an efficient detection solution.

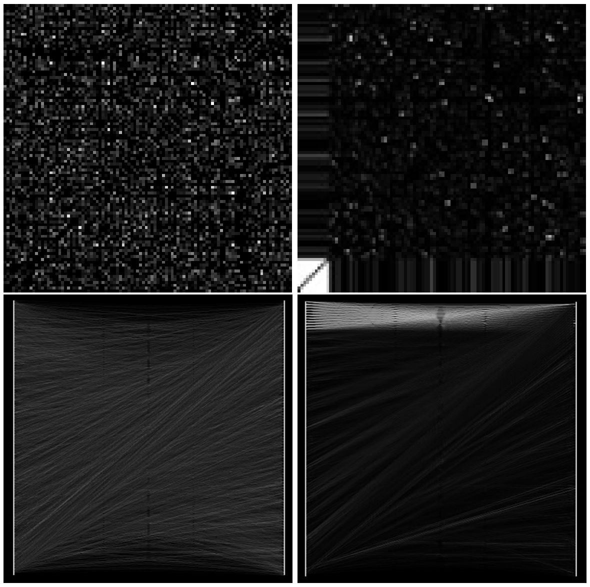


Fig. 7. Comparison of matrix visualization and parallel coordinates. Each column visualizes a topology matrix with the same color scheme.

While it is natural to use a matrix visualization approach to visualize topology information, other information visualization methods, such as parallel coordinates, can be used as well. We can use the row and column sequences as the orders of two axes in parallel coordinates to show highlighted band patterns among malicious nodes. As shown in Figure 7, we use the same scheme to color the matrices and parallel coordinates for comparison. Since we visualize statistical topology matrices, the node connections are dense, which are more challenging for parallel coordinates because of the line overlapping issue. On the left column, both visualizations do not show obvious patterns; while on the right column, both visualizations represent the highlighted region well, but parallel coordinates do not have the band structures in the matrix visualization. Also, since topology data is represented as matrices, matrix visualization is more intuitive for users, who are likely familiar with the matrix representation of topologies, to understand; thereby easier for them to identify malicious nodes. Therefore, we believe that matrix visualization is more appropriate for this application.

In this method, it is crucial to divide time steps reasonably; otherwise the detection accuracy of our method will be affected. As shown in the previous discussion, our method does not require accurate separation of attack and normal periods. As long as malicious activities occupy majority duration of a suspicious time period, our generated topology patterns can assist users to capture their subtle traces. However, when an attack only occupies a small portion of a time period, the attack patterns can be hidden in the normal patterns. This is possible under complicated attacks which involve the combination of multiple attack groups. For such cases, we expect that users may spend more time on adjusting suspicious time periods.

Compared to automated security algorithms, our approach does not rely on extra physical devices or any assumptions about the application scenarios. We generate network patterns purely based on features from information of accumulated network topologies. Also, with a small amount of user interaction, we can achieve a robust solution to detecting complicated Sybil attacks. Due to privacy issues, it is very hard to attain real networking data. Simulations are often used in the security community for evaluating automated approaches as documented in related work [6, 10, 13, 24, 21].

8.5 Quantitative Results

The running times of pattern generation and evaluation methods are very short, ranging from 0.00007 to 0.05006 seconds on average for 1000 time steps on a computer with 3.0GHz CPU and 2 GB RAM. Most of the interaction tools are in real time, so that users can interactively detect Sybil attacks.

9 Conclusions and Future Work

This paper presents a robust approach to detect Sybil attacks in wireless networks through analyzing statistical topology patterns. We characterize the attack features and detect malicious nodes with automatic pattern generation, evaluation, and interaction methods. Since we consider multiple relevant topology patterns, our method is robust to the detection of various complex attacks according to different aspects of attack features. Because of its flexibility, this design can be extended to provide a generalized detection solution. We have simulated real-life scenarios with different combinations of network parameters to test our approach and the results demonstrate that we can effectively identify various Sybil attacks. Since our approach explores hidden patterns in the network topology that will be impacted by many attacks on wireless networks, our approach has the potential to be applied to detecting other attacks.

The presented approach is composed of several closely related components. First, our pattern generation approaches can be viewed as matrix-based visualization methods, which have been used in a wide range of applications. Since we design automatic reordering algorithms according to attack features, they are more effective in exposing hidden matrix patterns than general heuristic algorithms. Second, we present automatic pattern evaluation methods by comparing matrix structures instead of data values, which is a new addition to matrix-based visualization methods. Third, a new time histogram and an automatic time segmentation method have been designed to provide useful visual cues for users to analyze time-series network data effectively. Fourth, we significantly simplify user interactions during multiple pattern visualizations through developing convenient interaction and analysis tools, thus the amount of user interaction in this approach is very limited.

We are interested in performing the following tasks to extend the capabilities of our integrated detection approach. First, we plan to design and perform user studies to evaluate how our system can be used by network administrators. Second, we plan to explore other methods to visualize our topology patterns based on their evaluation scores and human interaction. Third, we plan to develop efficient detection mechanisms for other attacks through extracting and identifying their topology pattern features. Our long term goal is to develop practical security systems that can perform multiple detection, analysis, and monitoring tasks for large scale networks in real life applications.

Acknowledgements

We wish to thank the editors and reviewers for their valuable comments. This research is supported by DOE DE-FG02-06ER25733, NSF 0633150, and NSF 0754592.

References

[1] G. R. Abuaitah and B. Wang. Secvizer: A security visualization tool for qualnet- generated traffic traces. VisSec poster, 2009.

[2] M. Ancona, W. Cazzola, S. Drago, and G. Quercini. Visualizing and managing network topologies via rectangular dualization. In *IEEE Symposium on Computers and Communications*, 2006.

[3] S. Au, C. Leckie, A. Parhar, and G. Wong. Efficient visualization of large routing topologies. *International Journal of Network Management*, 14(2):105–118, 2004.

[4] R. Ball, G. Fink, A. Rathi, S. Shah, and C. North. Home-centric visualization of network traffic for security administration. In *Proc. of ACM VizSEC/DMSEC*, 2004.

[5] R. A. Bazzi and G. Konjevod. On the establishment of distinct identities in overlay networks. In *Proceedings of ACM symposium on Principles of distributed computing*, pages 312–320, 2005.

[6] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *USENIX OSDI*, pages 299–314, 2002.

[7] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proc. of IEEE Symp. on S&P*, 2003.

[8] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *ACM Workshop on Economics of P2P systems*, pages 128–132, 2005.

[9] P. Doreian, V. Batagelj, and A. Ferligoj. *Generalized Blockmodeling (Structural Analysis in the Social Sciences)*. Cambridge Press, 2004.

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

[11] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key predistribution scheme for wireless sensor networks. In *Proc. of ACM CCS*, pages 42–51, 2003.

[12] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proc. of ACM CCS*, pages 41–47, 2002.

[13] Y. Fu, J. Chase, B. Chun, S. Schwab, and A. Vahdat. Sharp: an architecture for secure resource peering. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 133–148, 2003.

[14] N. Henry 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.

[15] B. Huffaker, E. Nemeth, and K. Claffy. Otter: A general-purpose network visualization tool. In *International Networking Conference (INET)*, 1999.

[16] W. jen Hsu, K. Merchant, H. wei Shu, C. hsin Hsu, and A. Helmy. Weighted waypoint mobility model and its impact on ad hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(1):59–63, 2005.

[17] C. Johnson and C. Hansen. *Visualization Handbook*. Academic Press, Inc., Orlando, FL, USA, 2004.

[18] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.

[19] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 79–86, 1998.

[20] Y. Koren and D. Harel. Multi-scale algorithm for the linear arrangement problem. Technical report, Technical Report MCS02-04, Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, 2002.

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

[22] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security*, 8(1):41–77, 2005.

[23] C. Muelder and K.-L. Ma. Visualization of sanitized email logs for spam analysis. In *Proceedings of APVIS*, 2007.

[24] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proc. of International Symposium on Information processing in sensor networks*, pages 259–268, 2004.

[25] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus context visualization for tabular information. In *Proc. ACM Conf. Human Factors in Computing Systems CHI*, 1994.

[26] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson. Idgraphs: Intrusion detection and analysis using stream compositing. *IEEE Comput. Graph. Appl.*, 26(2):28–39, 2006.

[27] H. Siirtola and E. Mäkinen. Constructing and reconstructing the reorderable matrix. *Information Visualization*, 4(1):32–48, 2005.

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

[29] W. Wang and A. Lu. Visualization assisted detection of sybil attacks in wireless networks. In *Proceedings of ACM Workshop on Visualization for Computer Security (VizSEC)*, pages 51–60, 2006.

[30] W. Yurcik. Visflowconnect-ip: A link-based visualization of netflows for security monitoring. In *18th Annual FIRST Conference on Computer Security Incident Handling*, 2006.