Practical Algorithms for Destabilizing Terrorist Networks

Nasrullah Memon and Henrik Legind Larsen

Software Intelligence Security Research Center,
Department of Software and Media Technology
Aalborg University Niels Bohrs Vej 8
6700 Esbjerg Denmark
{nasrullah, legind}@cs.aaue.dk

Abstract. This paper uses centrality measures from complex networks to discuss how to destabilize terrorist networks. We propose newly introduced algorithms for constructing hierarchy of covert networks, so that investigators can view the structure of terrorist networks / non-hierarchical organizations, in order to destabilize the adversaries. Based upon the degree centrality, eigenvector centrality, and dependence centrality measures, a method is proposed to construct the hierarchical structure of complex networks. It is tested on the September 11, 2001 terrorist network constructed by Valdis Krebs. In addition we also propose two new centrality measures i.e., position role index (which discovers various positions in the network, for example, leaders / gatekeepers and followers) and dependence centrality (which determines who is depending on whom in a network). The dependence centrality has a number of advantages including that this measure can assist law enforcement agencies in capturing / eradicating of node (terrorist) which may disrupt the maximum of the network.

1 Introduction

Many diverse systems in different research fields can be described as complex networks, that is, connecting the nodes together by the edges with nontrivial topological structures (Strogatz, S. H., 2001). Detailed works have been focused on several distinctive statistical properties sharing among a large amount of real world networks, to cite examples, the clustering effect (Albert, R. and Barabási, A.L., 2002, Dorogovtsev, S.N.; Mendes,J.F.F., 2002) and the right-skewed degree distribution (Albert, R. and Barabási, A.L., 1999). In this paper we consider another property sharing among many networks, the hierarchical structure of a complex network.

Hierarchy, as one common feature for many real world networks has attracted special attention in recent years (Ravasz, E., Barabási, A.L., 2003). In a network, there are usually some groups of nodes where the nodes in each group are highly interconnected with each other, while there are few or no links between the groups. These groups can induce a high degree of clustering, which can be measured with the connectivity probability for a pair of the neighbors of one node. This property coexists usually with the right-skewed degree distributions. The coexistence of these two properties tells us that the groups should combine in a hierarchical manner. Hierarchy is one of the key aspects of a theoretical model (Ravasz, E., Barabási, A.L., 2003) to

S. Mehrotra et al. (Eds.): ISI 2006, LNCS 3975, pp. 389 – 400, 2006.

capture the statistical characteristics of a large number of real networks, including some social networks (Newman, M. E. J., 2003).

As covert networks share some features with innocent individuals (overt networks), they are harder to identify because they mask their transactions. Another complicating factor is that covert / terrorist networks are often embedded in a much larger population. Hence, it is desirable to have tools to correctly classify individuals in covert networks so that the resources for destabilizing them will be used more efficiently.

To assist law enforcement and intelligence agencies to ascertain terrorist network knowledge efficiently and effectively, we proposed a framework of automated analysis, visualization and destabilization of terrorist networks (Memon, N. et al., 2004). Based on this framework, we developed a prototype called *iMiner* that incorporated several advanced techniques, for automatically detecting cells from a network, identifying various roles in a network (e.g., central members, gatekeepers, and followers), and may also assist law enforcement about the effect on the network after capturing a terrorist in a network.

The three innovative points of our paper are:

- The use of new measure Position Role Index (PRI) on the pattern of efficiency introduced by Vito Latora and Massimo Marchiori. This measure identifies leaders / gatekeepers and followers in the network. The algorithms for efficiency, importance of critical nodes in a network and PRI are also presented.
- The use of another measure known as Dependence Centrality (DC) which discovers who is depending on whom in a network. The algorithm of DC is also presented.
- Estimate possible hierarchical structure of a complex network by applying degree centrality and Eigenvector centrality from social network analysis (SNA) literature and combining it with new measure dependence centrality. The algorithm for estimating the possible hierarchical structure of the terrorist network is also shown. The all the algorithms presented in the paper are designed and developed by the authors.

The remainder of the paper is organized as follows: Section 2 briefly describes the motivation of this research and existing destabilizing approaches for terrorist networks; Section 3 describes fundamentals of networks analysis; whereas Section 4 discusses algorithms and techniques for destabilizing terrorist networks. Section 5 shows how hierarchy is constructed from covert networks and Section 6 concludes the paper.

2 Motivation

When intelligence agencies arrest a few members of a terrorist cell, how can they know if the cell has been disabled?

Social scientists have imagined individual terrorists as nodes on a graph, most of them are connected to only one or two other nodes. Using such cellular graphs, researchers have proposed ways of estimating whether a chain of relationships has been effectively shattered, even when some of its members elude capture.

There is a growing amount of literature on modeling terrorist networks as graphs, an outgrowth of the existing literature concerning other types of criminal networks (Krebs, V., 2002, Klerks, P., 2001). There is also a small amount of literature on destabilizing networks, modeled as graphs, by seeing how connections do or do not dissipate when nodes are removed (Carley, K. M., Lee, J-S. and Krackhardt, D., 2002; Carley, K. M. et al., 2003).

A graph model, however, may not be the best one available for representing a typical terrorist organization (Farley, J. D., 2003). His views are that modeling terrorist networks as graphs does not give us enough information to deal with the threat. Lattice theory is the abstract study of order and hierarchy. In terrorist organizations, hierarchy appears to matter. "Modeling terrorist cells as graphs ignores an important aspect of their structure, namely their hierarchy, and the fact that they are composed of leaders and followers" (Farley, J. D., 2003).

We have related the concept of hierarchy and graph and predicted the structure of a non hierarchical network so that it can be viewed as a hierarchy. Our results for September 11 terrorist network (Krebs, V., 2002), are in excellent agreement to reality.

3 Social Network Analysis (SNA)

In general, the network studied in this paper can be represented by an undirected and un-weighted graph G = (V, E), where V is the set of vertices (or nodes) and E is the set of edges (or links). Each edge connects exactly one pair of vertices, and a vertex pair can be connected by (a maximum of) one edge, i.e., multi-connection is not allowed.

A terrorist network consists of V set of actors (nodes) and E relations (ties or edges) between these actors. The nodes may be individuals, groups (terrorist cells), organizations, or terrorist camps. The ties may fall within a level of analysis (e.g. individual to individual ties) or may cross levels of analysis (individual-to-group analysis). A terrorist network can change in its nodes, links, groups, and even the overall structure. In this paper, we focus on detection and description of node level dynamics.

Mathematically, a network can be represented by a matrix called the *adjacency matrix* **A**, which in the simplest case is an *n x n* symmetric matrix, where *n* is the number of vertices in the network. The adjacency matrix has elements.

A_{ij} = 1, if there is an edge between vertices i and j, and 0 otherwise.

The matrix is symmetric since if there is an edge between *i* and *j* then clearly there is also an edge between *j* and *i*. Thus $A_{ij} = A_{ji}$.

3.1 Node Level Measures

As terrorists establish new relations or break existing relations with others, their position roles, and power may change accordingly. These node dynamics resulting

from relation changes can be captured by a set of centrality measures from SNA. The centrality measures address the question, "Who is the most important or central person in the network?"

There are many answers to this question, depending on what we mean by important. Perhaps the simplest of centrality measures is *degree centrality*, also called simply *degree*. The degree of a vertex in a network is the number of edges attached to it. In mathematical terms, the degree k_i of a vertex *i* is (Newman, M. E. J., 2003):

$$k_i = \sum_{j=1}^{n} A_{ij} \tag{1}$$

Though simple, *degree* is often a highly effective measure of the influence or importance of a node: in many social settings people with more connections tend to have more power.

A more sophisticated version of the same idea is the so-called *eigenvector centrality*. Where degree centrality gives a simple count of the number of connections a vertex has, eigenvector centrality acknowledges that not all connections are equal. If we denote the centrality of vertex *i* by x_i, then we can allow for this effect by making x_i proportional to the average of the centralities of *i*'s network neighbors (Newman, M. E. J., 2003):

$$x_i = \frac{1}{\lambda} \sum_{i=1}^{n} A_{ij} x_j \tag{2}$$

where λ is a constant. Defining the vector of centralities **x** = (*x*1; *x*2; : : :), we can rewrite this equation in matrix form as:

$$\lambda x = \mathrm{A} \bullet x \tag{3}$$

Hence we see that **x** is an eigenvector of the adjacency matrix with eigenvalue λ. Assuming that we wish the centralities to be non-negative, it can be shown that λ must be the largest eigenvalue of the adjacency matrix and **x** the corresponding eigenvector.

4 Destabilizing Terrorist Networks

4.1 The Efficiency E(G) of a Network

The network efficiency E (G) is a measure to quantify how efficiently the nodes of the network exchange information (Latora, V., Marchiori, M., 2004). To define efficiency of G first we calculate the shortest path lengths $\{d_{ij}\}$ between two generic points *i* and *j*. Let us now suppose that every vertex sends information along the network, through its edges. The efficiency ε_{ij} in the communication between vertex i and j is inversely proportional to the shortest distance: $\varepsilon_{ij} = 1/d_{ij} \ \forall$ i, *j* when there is no path in the graph between i, and *j*, we get $d_{ij} = +\infty$ and consistently $\varepsilon_{ij} = 0$. N is known as the size of the

network or the numbers of nodes in the graph. Consequently the average efficiency of the graph of G can be defined as (Latora V., Marchiori, M., 2004):

$$E(G) = \frac{\sum_{i \neq j \in G} \varepsilon_{ij}}{N(N-1)} = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}} \quad (4)$$

The above formula gives a value of E that can vary in the range [0, ∞], while it be more practical to normalize E in the interval of [0, 1].

4.2 The Critical Components of a Network

Latora V. et al recently proposed a method to determine network critical components based on the efficiency of the network briefly discussed in the previous subsection. This method focuses on the determination of the critical nodes. The general theory and all the details can be found in Ref. (Latora V., Marchiori, M., 2004).

The main idea is to use as a measure of the centrality of a node i the drop in the network efficiency caused by deactivation of the node. The importance I ($node_i$) of the $_i$th node of the graph G is therefore:

$$I(node_i) \equiv \Delta E = E(G) - E(G - node_i), \; i = 1,...,N, \quad (5)$$

Where $G - node_i$ indicates the network obtained by deactivating $node_i$ in the graph G. The most important nodes, i.e. the critical nodes are the ones causing the highest ΔE. The results of deactivation of nodes for 9-11 hijackers and their affiliates are shown in figure 1 and figure 2.

4.3 Position Role Index (PRI)

The PRI is our proposed measure which highlights a clear distinction between followers and gatekeepers (It is a fact that leaders may act as gatekeepers). It depends on the basic definition of efficiency as discussed in equation (4). It is also a fact that the efficiency of a network in presence of followers is low in comparison to their absence in the network. This is because they are usually less connected nodes and their presence increases the number of low connected nodes in a network, thus decreasing its efficiency.

If we plot the values on the graph, the nodes which are plotted below x-axis are followers, whereas the nodes higher than remaining nodes with higher values on positive y axis are the gatekeepers. While the nodes which are on the x-axis usually central nodes, which can easily bear the loss of any node. The leaders tend to hide on x-axis there.

We applied this measure on the network of alleged 9-11 hijackers, (Krebs, V., 2002) and results are shown in figure 1. The algorithms for PRI, efficiency of network and critical components of network are described in Exhibit 1.

Exhibit 1. Algorithms for Efficiency, Delta Efficiency and Position Role Index

Algorithm for Efficiency E(G)

Let G be a graph, N is a set of nodes which are contained by G and E is the set of edges through which the nodes of graph are connected. Let m is the number of elements in N

```
Input:
Graph G, N set of its nodes
Output:
Efficiency of Graph G
Let s=0 and e=0
For each element n1 in N
        For each element n2 in N
        Let s = s + 1/ d (n1, n2)
        Next n1
        Let e= e + s
Next n2
Let e=1/ (m * (m– 1)) * e
Return e
```

Where "e" is the efficiency of the graph and d(n1, n2) is the function which gives us the distance of shortest path from n1 to n2.

Algorithm for Finding Delta Efficiency:

Suppose n is the node for which we are finding delta efficiency. Let G' be a subgraph similar to G, only the difference is that E is the set of edges through which the nodes of graph are connected except the edges which originate or point to n. (G' does not contain any edge to or from n). Let N is a set of Nodes in G' and m is the number of elements in N.

```
Input:
Graph G, N set of its nodes, n is the node for which we are finding the value of Delta Efficiency and remove edges to or from n in G to get G'
Output:
Delta Efficiency of n in G.
Let s=0 and de=0
For each element n1 in N
        For each element n2 in N
        Let s = s + 1/ d (n1, n2)
        Next n1
        Let de= de + s
Next n2
Let de=1/ (m * (m – 1)) * de
Let efficiency=E (G)
Let de = ((efficiency – de) / efficiency)
Return de
```

Algorithm for Finding Position Role Index

Suppose n is the node for which we are finding delta efficiency. Let G' be the subgraph that is similar to G – n1. (G' does not contain any edge to or from n and also it does not contain n). Let N is a set of Nodes in G' and m is the number of elements in N.

```
Input:
Graph G, N set of its nodes, n is the node for which we are finding the value of NI and remove n and all the edges coming from or to n in G to get G'
Output:
NI of n in G.
Let s=0 and ni=0
For each element n1 in N
        ni=0
        For each element n2 in N
          Let s = s + 1/ d (n1, n2)
        Next n1
        Let ni= ni + s
Next n2
Let ni = 1/ (m * (m– 1)) * ni
Let efficiency = E (G)
Let ni=((efficiency – ni)/efficiency)
Return ni
```

4.4 Dependence Centrality (DC)

The DC is the recently introduced measure by the authors (Memon, N., Legind, H.L., 2006). The dependence centrality of a node is defined as how much that node is dependent on any other node in the network. Mathematically it can be written as:

$$DC_{mn} = \sum_{m \neq p,\, p \in G} \frac{d_{mn}}{N_p} + \Omega \tag{6}$$

Where **m** is the root node which depends on **n** by $\mathbf{DC_{mn}}$ centrality and $\mathbf{N_p}$ actually is the Number of geodesic paths coming from **m** to **p** through **n**, and $\mathbf{d_{mn}}$ is geodesic distance from ***m*** to ***n***. The Ω is taken 1 if graph is connected and 0 in case it is disconnected. In this paper we take Ω as 1, because we consider that graph is connected. The first part of the formula tells us that:

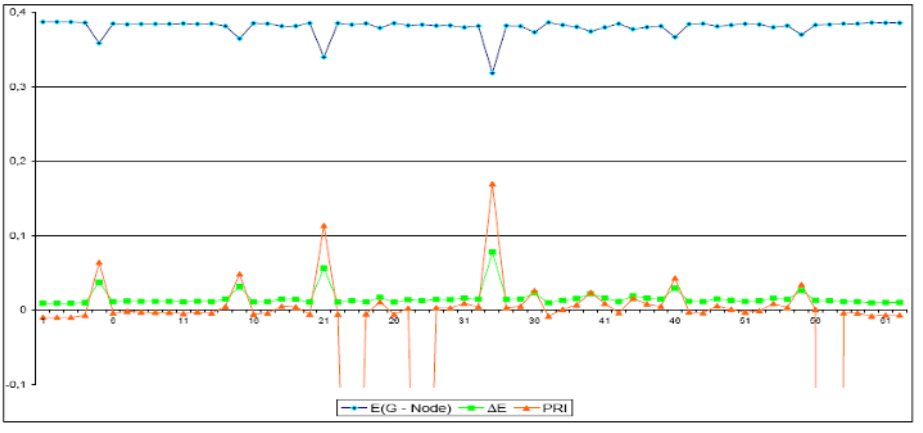

Fig. 1. The efficiency of the original network is E (G) = 0.395. The removed node is shown on x-axis; the efficiency of the graph once the node is removed is reported as E ($G - Node_i$), while the importance of the node (drop of efficiency) is shown as Δ E. While position role index shown as PRI of the removed node. The results prove important aspects of the network and confirmed that Mohammed Atta (node # 33) was the ring leader.

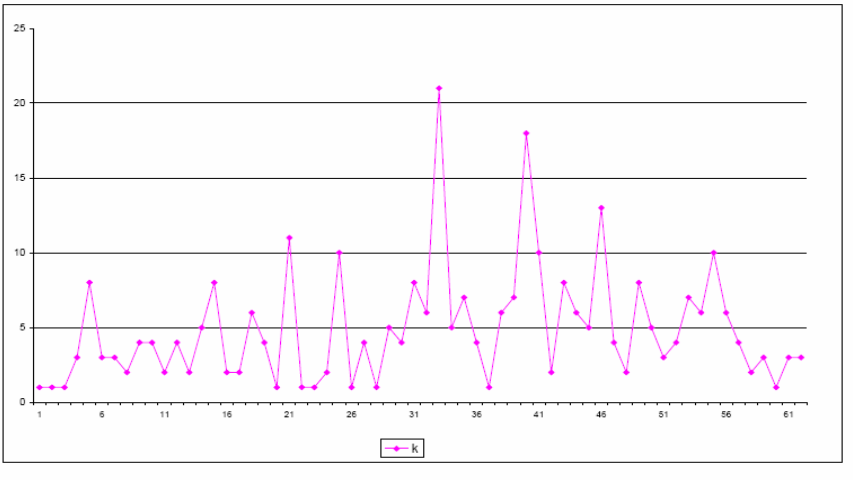

Fig. 2. An alternative measure of the importance of the node (k), the degree of (i.e. the number of links incident with) the removed node

How many times **m** uses **n** to communicate other node **p** of the network? In simple words **p** is every node of the network, to which **m** is connected through **n** (The connection represents the shortest path of node **m** to **p**, and **n** is in between). $\mathbf{N_p}$ represents the number of alternatives available to **m** to communicate to **p** and $\mathbf{d_{mn}}$ is the multiplicative inverse of geodesic distance (**1/d**).

Algorithm for Dependence Centrality

For each node "*n*" in graph
  For each node "n_d" in graph other than "n"
    Compute All shortest paths from "*n*" to "n_d", and save it in vector "*paths*"
    For each path "*p*" in "*paths*"
      For each node "*nt*" in p
        - Compute the Shortest distance "*d*" from node "n" to "*nt*".
        - Find multiplicative inverse 1/"*d*" of shortest distance from "*n*" to "n_t".
        - Divide it with the number of paths in vector "*paths*". let the result be "D_t ".
        - Retrieve the value of "D_{mn}" from HashMap "map_t" corresponding to "n_t", and add it with "D_t" + 1. Then save the result again in HashMap "map_t" against the key "n_t".

      Next
    Next
  Next
Next

This measure shows that how much node **m** is dependent on the node **n**. We can also say that how much node **n** is useful to node **m** in order to communicate with other nodes of the network.

The node which has less in summation of *dependence centrality might be key player* i.e., leader / gatekeeper, who usually direct many other peripherals and control communication. The key players have low *dependence centrality* (DC) as they have large number of direct links with other nodes of the network and they do not depend on others to communicate with those nodes.

When we tabulate the Dependence Centrality (DC), a matrix is obtained; where each row corresponds to a particular node, its DC against all the nodes are represented in the form of values ($1 \leq$ values $\geq$ (total nodes – 1)) at different columns in the same row. When we sum up all of these values in a row, the sum shows how much the node is dependent on other nodes. The lower the sum, the less will be the node *dependent* on other nodes or that node is said to be an *independent* node. Similarly if we sum each column, it will show how much all the nodes depend on that particular node which is associated with that column. The dependence centralities of the hijackers and their affiliates as shown in figure 3 can be seen at http://cs.aue.aau.dk/~nasrullah/DC_9_11.htm. There are some interesting results from this 62x62 matrix that shows, if *node 33* is removed, *nodes 23 and 28* will also be *isolated completely* from the network. Similarly if *node 38* is eradicated/captured, *nodes 20, 22, and 26* are *totally isolated* from the network. The rationale behind that is the nodes (for example, 23, 28; 20, 22, 26) are

completely depending on the nodes (for example, 33 and 38 respectively). If the nodes are completely depending on the other nodes, they will be isolated (cut-off from the network completely) by capturing the node on which those nodes are depending.

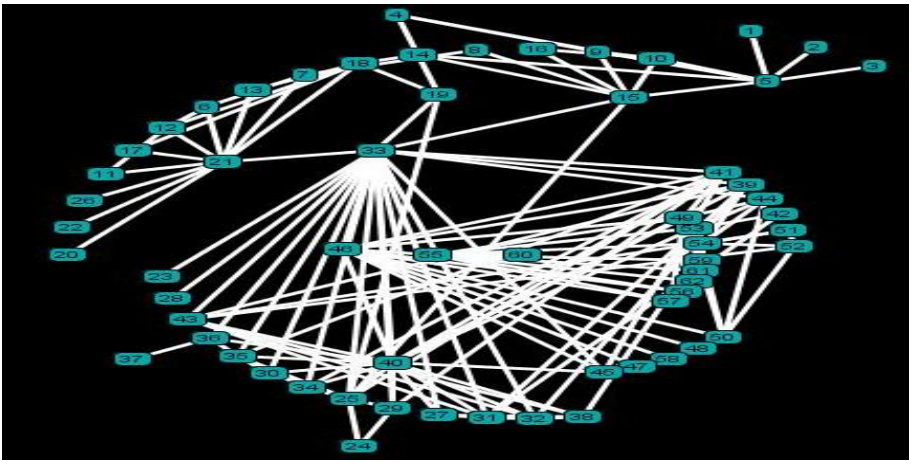

Fig. 3. 9/11 hijackers and their affiliates dataset. The names of terrorists shown in Appendix A at the end of paper.

5 Construction of Hierarchy for 9-11 Terrorists' Network

By using algorithms shown in Exhibit 2, we have constructed the *hierarchy* shown in figure 4 (using *iMiner*), of the hijackers involved in 9/11 terrorist attack and their affiliates (from the publicly available dataset as shown in figure 3).

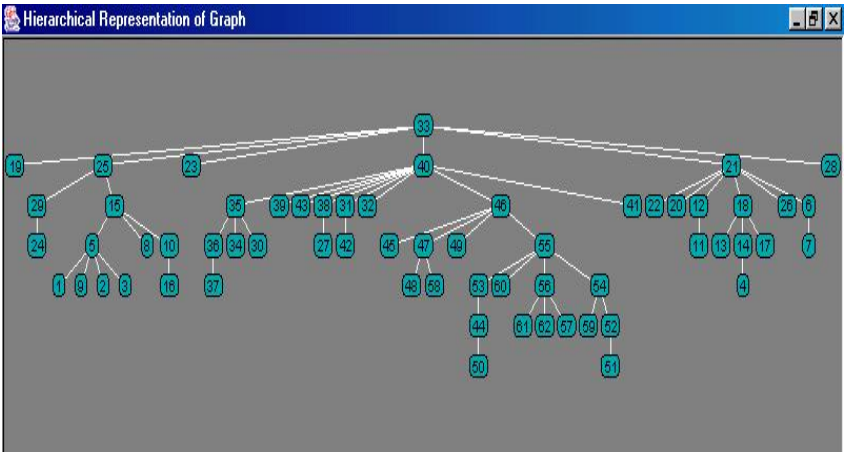

Fig. 4. Hierarchy discovered by iMiner from graph shown in figure 3, using Algorithms shown in Exhibit 2

The hierarchy clearly suggests that *Muhammad Atta* (33) was the key leader of the plot. While *Marvan Al Shehhi* (40) was assisting him as he is below in the hierarchy. They both were suggested as potential leaders in 9/11 attack and led their respective groups. They were also both members of Hamburg Cell. *Fayez Ahmed* (31), and *Mohand Al Shehhi* (42), who were in the same hijacked plane with *Marvan Al Shehhi,* are below Marvan Al Shehhi. While *Abdul Aziz Alomari* (39), *Waleed Al Shehhi* (38) are in 3rd level in the hierarchy.

The intelligence agencies can easily detect who are potential leaders / gatekeepers and even peripheries by using these new algorithms.

Exhibit 2. Algorithms for Constructing Hierarchy

Algorithm. Converting undirected graph G into directed D

1. Take any node "n" of graph G, and find its neighbors "N".
2. Take a node "s" such that s ∈ N (N is set of neighbors of n.). Compare Degree Centrality of s to Degree Centrality of n,
    - if Degree Centrality of s > Degree Centrality of n, Mark a directed edge from s to n.
    - if Degree Centrality of s < Degree Centrality of n, Mark a directed edge from n to s.
    - if Degree Centrality of s = Degree Centrality of n
        1. Compare Eigen-Vector Centrality of s to Eigen-Vector Centrality of n,
            - If Eigen-Vector Centrality of s > Eigen-Vector Centrality of n, Mark a directed edge from s to n.
            - If Eigen-Vector Centrality of s < Eigen-Vector Centrality of n, Mark a directed edge from s to n.
            - If Eigen-Vector Centrality of s = Eigen-Vector Centrality of n, Ignore the link.
3. Repeat Step 2 for every member of N.
4. Repeat Step 1 for every node of graph G.

Algorithm. To make Tree T from Directed Graph D

1. Take any node "n" of directed Graph "D", and find all the nodes "N(n)" adjacent to edges originating from node n. and mark them as Children of n. Here N(n) is neighbors N of node n.
2. Find all the nodes (parents) "P" adjacent to edges pointing to node n and mark them as Parents of n.
3. Repeat step 1 and 2 for all nodes of Directed Graph D.
4. Again take any node "n" of directed Graph "D",
5. If number of elements in P (where P is the set of Parents of n) is 0, then add "root "of Tree "T" as its parent and mark node n as children of "root".
6. If number of elements in P > 1, Remove all the nodes except "p1" from P, such that (N (p1) ∩ N (n)) is maximum.(Where N (p1) is the set of Neighbors of p1). Also mark n as Children of p1.

7. If number of elements in P is still > 1, remove all the nodes from P except the node p1, for which the n has highest Dependence Centrality. Also mark n as Children of p1.
8. If number of elements in P is still > 1, Remove all of its parents and then add "root" of Tree T as its parent and also mark node n as children of "root".
9. Repeat Step 4 to 8, for all nodes of directed graph D.
10. Draw Tree T.

6 Conclusions

In this paper we have proposed new practical algorithms which can assist law enforcement agencies to discover who is under the influence of whom in a network by visualizing the hierarchal chart. The position role index measure assists in finding about who is who in a network. The dependence centrality determines which individuals are depending on which nodes, in order to help investigators to disrupt the network. All the algorithms discussed in the paper are implemented in the prototype *iMiner*. The prototype can provide assistance to law enforcement agencies, indicating when the *capture* of a *specific* terrorist will likely disrupt the terrorist network. Moreover, using *iMiner* an investigator has the power to estimate the network's size, determine its membership structure, find who the most important terrorist in the network is, determine the efficiency of the network, unearth the leaders / gatekeepers / followers, and determine on which node the maximum nodes in the network depends.

References

1. Albert, R. and A.L. Barabási, (2002) *Dynamics of complex systems: scaling laws for the period of boolean networks*, Physics Reviews. **47**
2. Albert, R. and A.L. Barabási, (1999) *Emergence of scaling in random networks*, Science, **286**:509-512
3. Carley, Kathleen M., Lee Ju-Sung, Krackhardt, D. (2002) Destabilizing Networks. *Connections* **24** (3) 79-92
4. Carley, K. M. et al. (2003) Destabilizing Dynamic Covert Networks. In *Proceedings of 8th International Command and Control Research and Technology Symposium.* Conference held at National Defense War College, Washington, DC. Evidence Based Research Vienna, VA.
5. Dorogovtsev, S.N. and J.F.F. Mendes, (2002) Evolution of Networks *Adv. Phys*. **51**, 1079
6. Farely, David J. (2003) Breaking Al Qaeda Cells: A Mathematical analysis of counterterrorism *Operations Studies in conflict terrorism.* **26**:399–411
7. Klerks, P. (2001) The network paradigm applied to criminal organizations", *Connections* **24** (3)
8. Krebs, V. (2002) Mapping Terrorist Networks, *Connections* **24**(3*)*
9. Latora, V., Massimo Marchiori (2004) How Science of Complex Networks can help in developing Strategy against Terrorism, *Chaos, Solitons and Fractals* **20**, 69-75
10. Memon Nasrullah, Daniel Ortiz Arroyo, Henrik Legind Larsen (2004) *Investigative* Data Mining: A General Framework. In *Proceedings* of *International Conference on Computational Intelligence*, Istanbul, Turkey 384-387
11. Memon Nasrullah, Henrik Legind Larsen (2006) Practical Approaches for Analysis, Visualization and Destabilizing Terrorist Networks. In *Proceedings of ARES 2006: The First*

International Conference on Availability, Reliability and Security, Vienna University of Technology Austria.
12. Newman, M. E. J. (2003) The structure and function of complex networks, *SIAM Review* **45**, 167- 256
13. Ravasz, E. A.L. Barabási, (2003) Hierarchical organization in complex networks, *Phys. Rev. E*, **67**, 261121
14. Strogatz, S. H. (2002) Exploring Complex Networks, *Nature 410*, 268-276
15. Trusina, A. S.; Maslov, P. Minnhagen and K. Sneppen, (2004) Hierarchy and Anti-Hierarchy in Real and Scale Free Networks *Phys. Rev. Lett*. **92**, 178702

Appendix A. Names of 9/11 Hijackers and Their Affiliates

1. Jean Marc Grandvisir
2. Abu Zubaida
3. Nizar Trabelsi
4. Abu Walid
5. Djamal Beghal
6. Mehdi Khammoun
7. Hyder Abu Doha
8. Ahmed Ressam
9. Kamel Dauod
10. Jerome Courtailler
11. Lased Ben Heni
12. Mohamed Bensakhria
13. Essoussi Laaroussi
14. Abu Qatada
15. Zaoarias Moussaoui
16. David Courtailler
17. Seifallah ben hassine
18. Tarek Maaroufi
19. Imdad Eddin Barakat Yarkas
20. Fahid al sharki
21. Essid Sami Ben Khemais
22. Madjid Sahoune
23. Abdelghani Mzoudi
24. Mohamed Belfas
25. Ramzi Bin al Shibh
26. Samir Kishk
27. Mustafa Ahmed Al Hissaw
28. Ahmed Khalil Ibrahim Samir
29. Agus Badim
30. Mounir el motassaeq
31. Fayez Ahmed
32. Wail Al Shehhi
33. Mohamed Atta
34. Zakariya Essabar
35. Said Bahaji
36. Mamoun Darkazanali
37. Mamduh Mahmud Salim
38. Waleed Al Shehhi
39. Abdul Aziz Al Omari
40. Marwan Al Shehhi
41. Ziad Jarrah
42. Mohand Al Shehhi
43. Satam Suqami
44. Ahmed Al Haznawi
45. Lotfi Raissi
46. Hani Hunjor
47. Rayed Mohammed Abdullah
48. Bandar Alhazmi
49. Salem Alhazmi
50. Ahmed Alghamdi
51. Raed Hijazi
52. Nabil al Marabh
53. Hamza Alghamdi
54. Saeed Alghamdi
55. Nawaf Al hazmi
56. Khalid Al Mindhar
57. Majed Moqed
58. Faisal al salmi
59. Ahmed alnami
60. Mohamed Abidi
61. Abdussattar Shaikh
62. Ossama Awadallah