Using the Spanning Tree of a Criminal Network for Identifying its Leaders

Kamal Taha, *Senior Member, IEEE* and Paul D. Yoo, *Senior Member, IEEE*

***Abstract*— We introduce a forensic analysis system called ECLfinder that identifies the influential members of a criminal organization as well as the immediate leaders of a given list of lower-level criminals. Criminal investigators usually seek to identify the influential members of criminal organizations, because eliminating them is most likely to hinder and disrupt the operations of these organizations and put them out of business. First, ECLfinder constructs a network representing a criminal organization from either Mobile Communication Data associated with the organization or crime incident reports that include information about the organization. It then constructs a Minimum Spanning Tree (MST) of the network. It identifies the influential members of a criminal organization by determining the important vertices in the network representing the organization, using the concept of existence dependency. Each vertex *v* is assigned a score, which is the number of other vertices, whose existence in MST is dependent on *v*. Vertices are ranked based on their scores. Criminals represented by the top ranked vertices are considered the influential members of the criminal organization represented by the network. We evaluated the quality of ECLfinder by comparing it experimentally with three other systems. Results showed marked improvement.**

***Index Terms*—Forensic investigation, digital forensic, social network, criminal network, mobile communication data.**

I. Introduction

Social groups and their relationships have long been identified using Social network analysis (SNA) [2, 21, 22]. Inspired by SNA, researchers in digital forensic investigation have been employing similar network analysis techniques for identifying criminal communities, their relationships, and their influential leaders [12]. As a result, digital forensic has emerged as an important tool for investigation crimes. Usually, forensic investigators study and analyze communication records for the purpose of identifying criminal communities and their leaders. Recently, forensic investigators have shown a growing interest on using Mobile Communication Data (MCD) that belong criminal organizations to construct networks that depict the organizations and analyze these networks [12].

The interest on constructing networks from MCD came from the fact that most criminals involved in organized crimes (such as terrorism, drug trafficking, and criminal gangs) plot and contemplate their criminal activities through mobile phone communications [12]. Criminal forensic investigators analyze such networks to infer useful information such as: (1) the structure of the criminal organization, (2) the relationships between the criminals, (3) the influential members of the criminal organization, and (4) the flow of communications between the criminals. Recently, criminal forensic investigators have also shown interest on constructing networks from Crime Incident Reports that contain information about a criminal organization [6].

We propose in this paper a forensic analysis system called **ECLfinder** (**E**fficient **C**riminal **L**eaders **F**inder). ECLfinder can identify the most influential members of a criminal organization. Given a list of lower-level criminals in a criminal organization, ECLfinder can also identify the immediate leaders of these lower-level criminals. Identifying the influential members of a criminal organization is one of the most important tasks that criminal investigators undertake. Usually, members of a criminal organization, who hold central positions in a criminal organization, are targeted by criminal investigators for removal or surveillance [4, 15]. This is because these central members usually play key and influential roles in the organization by acting as commanders who issue instructions to other members or serve as gatekeepers, who receive and distribute information and goods to other members. Removing these central members is most likely to disrupt the organization and put it out of business.

Shang et al. [18] stated that a common problem in a criminal investigation involves a criminal organization is to identify the leaders of the organization. Memon [16] stated that the identification of key actor(s) in criminal covert networks is a major objective for criminal investigators and eliminating these key actors can destabilize the criminal network. Wiil et al. [25] stated that the identification and elimination of key nodes in a terrorist network would decrease the ability of the network to function normally.

In the framework of ECLfinder, a network can be constructed from either Mobile Communication Data (MCD) that belongs to a criminal organization or from crime incident reports that contain information about a criminal organization. A vertex in a network represents an individual and an edge represents the relationship between two individuals. First, ECLfinder constructs the *Minimum Spanning Tree* (MST) of the network. ECLfinder identifies the influential members of a criminal organization by determining the important vertices in the network, using the concept of existence dependency. It employs this concept to identify for each vertex *v*, the set *S* of vertices, whose existence in MST is dependent on *v*. This is because, if the existence of *S* in MST is dependent on *v*, *v* is

K. Taha is with the Electrical and Computer Engineering Department, Khalifa University, UAE (e-mail: kamal.taha@kustar.ac.ae).

P. Yoo is with Cranfield University and Defence Academy of United Kingdom, UK (email: paul.d.yoo@ieee.org).

influential to *S*. It then assigns a score to each vertex *v*, which is the number of vertices in the set *S*. Vertices are ranked based on their scores. Criminals represented by the top ranked vertices are considered the influential members of the criminal organization.

II. Background and Outline of the Approach

A. Background

A number of methods have been proposed for identifying the set of suspicious source nodes (e.g., fake followers, botnets, etc.) on a given criminal network. The authors of [1] investigated the network structure of Mafia syndicates by building two networks representing Mafia gangs operating in the North of Sicily. In the networks, a vertex represents an individual and an edge connecting two vertices represents the existence of at least one reciprocated phone call between the individuals associated with these vertices. The following are the objectives of the authors of [1]: (1) to understand the functional roles of the members of the Mafia syndicates, (2) to quantify the ability of a Mafia syndicate to react to police operations after the detention of some of its members, and (3) the resilience of Mafia syndicate to disruption caused by police operations.

The authors of [10] presented *LogViewer*, a Web-based criminal network analysis framework to study combinations of geo-embedded and time-varying data sources like mobile phone networks and social graphs. *LogViewer* aims at: (1) identifying criminal behaviors and uncovering illicit activities, (2) investigating the centrality of vertices representing criminals, (3) studying the flow of information over time, and (4) determining the physical closeness effects on networks. In 2013, Catanese et al. [7] introduced an initial version of a system called *LogAnalysis*. In this initial version, the system was intended for forensic visual statistical analysis of mobile phone logs. The system helps in understanding the hierarchies within criminal organizations and discovering key and central members inside the organizations [7].

Despite the success of most current methods for identifying the vertices that are important to query vertices, these methods suffer incomplete contribution and inconsistent contribution. Incomplete contribution occurs, if some query vertices do not contribute to the overall relative importance value of a vertex. The inconsistent contribution occurs, if query vertices contribute unequally to the overall relative importance value of a vertex. Let *v* be the current vertex under consideration. ECLfinder overcomes the problem of Incomplete Contribution by: (1) considering the importance of *each* query vertex to *v*, and (2) assigning a weight to each incoming edge to *v* that is outgoing from one of the query vertices (this weight represents the importance/rank of this vertex relative to all incoming edges to *v*). ECLfinder overcomes the problem of Inconsistent Contribution by: (1) considering the importance of *each* query vertex to each vertex connected to *v*, and (2) accounting for the degree of relativity of *v* to all query vertices.

B. Outline of the Approach

We present below an overview of our approach in terms of the *sequential* processing steps taken by ECLfinder to identify the influential members of a criminal organization.

1) ***Constructing a network:*** A network is constructed from either MCD associated with a criminal organization or crime incident reports that contain information about the members of a criminal organization.
2) ***Assigning a weight to each edge in the network:*** In a network constructed from MCD, the weight of an edge represents the number of phone calls/messages between two criminals. In a network constructed from crime incident reports, the weight of an edge represents the number of co-occurrences of the names of suspects and accomplices in the same reports.
3) ***Computing the shortest-path edge betweenness:*** We compute the "shortest-path edge betweenness" [11] for each edge based on the initial weights described in step 2. We replace edges' initial weights by their shortest-path betweenness.
4) ***Assigning a score to each edge:*** Edges' shortest-path betweenness are replaced by their inverses. This is because we will construct the MST of the network, which spans all vertices with the minimal sum of weights. The inverses are used as the scores of the edges.
5) ***Assigning a score to each vertex in the network based on the concept of existence dependency:*** We construct the MST of the network based on the edges' scores described in step 4. ECLfinder assigns a score to each vertex *v* in the network. The score of the vertex *v* is the number of other vertices, whose existence in the MST is dependent on *v*. The score represents the relative rank (i.e., importance) of the criminal represented by the vertex *v* in the criminal organization.
6) ***Identifying the influential members of the criminal organization:*** Vertices are ranked based on their scores described in step 5. Criminals represented by the top ranked vertices are considered the influential members of the criminal organization.

III. Constructing a Network

In the framework of ECLfinder, a network can be constructed from information gathered from MCD associated with a criminal organization. A vertex in such a network represents a criminal caller and/or receiver. An edge represents the flow of communications between two criminals, through phone calls or messages. The weight of an edge represents the number of phone calls/messages between the two criminals represented by the two vertices connected by the edge.

In the framework of ECLfinder, a network can also be created from crime incident reports that contain information about the members of a criminal organization. In such a network, a vertex represents a criminal. An edge represents the relationship between two criminals, determined based on the co-occurrences of the criminals' names in the same crime incident reports. ECLfinder employs the concept of space approach [5] to construct networks automatically from crime incident reports [6]. ECLfinder employs the techniques of Stanford Named Entity Recognition [17] to determine the names of people in reports. It uses a tokenizer and stemmer to match a sequence of words against persons' names. Let *n* be the number of co-occurrences of the names of two suspects (or accomplices) in the same crime incident reports. *n* is transformed into similarity weight for the edge connecting the two vertices in the network representing the two suspects.

We compute the "shortest-path edge betweenness" [11] for each edge based on the initial weights described above. We adopt the concept of edge betweenness proposed by Girvan–Newman [11]. We consider the shortest-path betweenness of an edge as the actual weight of the edge. Therefore, we replace edges' initial weights by their shortest-path betweenness. This is because the shortest-path betweenness of edges reflect the relative degree of relatedness between vertices better than the number of phone calls/messages between the vertices (or the co-occurrences of names in reports). The shortest-path edge betweenness computes the fraction of shortest paths passing through an edge. It can measure the rate at which information passes along each edge. Eventually, the weight of each edge is represented by the edge's shortest-path betweenness.

Finally, a *score* is assigned to each edge. The score of an edge is the inverse of the edge's shortest-path betweenness weight. Therefore, a smaller score represents a stronger relationship between the two vertices connected by the edge. That is, the smaller the score of an edge the closer the relationship between the two vertices connected by the edge. We adopt this approach in order to construct the MST of the network. This is because the path of the MST spans all vertices with the *minimal sum of weights*: the MST's path has the smallest sum of the weights of edges connecting all vertices compared to all other paths that span all the vertices.

IV. Identifying the Influential Members of a Criminal Organization

A. Assigning a Score to each Vertex in the Network based on the Concept of Existence Dependency

We construct the Minimum Spanning Tree (MST) of the network based on the edges' scores described in section III. A MST is a tree that spans all the vertices of a network and the sum of the scores of the edges connecting the vertices is the smallest among all other trees that span all the vertices. We construct the MST, because its path spans all *closely related* vertices *(i.e., its path connects the vertices that represent the criminals, who have the highest degree of relationships)*. Usually, a criminal organization is hierarchically structured in terms of power. The MST can represent the path of passing information through this hierarchical structure.

Algorithm *CONSTR-MST* in Fig. 1 constructs an MST based on Prim's algorithm. Let *V* be the set of vertices in the network. Let *MST* be a set that stores the edges of the tree. At each step, an edge with a light score is added to the current tree *MST* that connects the tree to an isolated vertex. The inputs to the algorithm are a network *NW*, the scores of the edges *S* (recall section III), and the root vertex *r*. In line 3 of the algorithm, the parent of each vertex *u* is assigned to NIL and stored in variable π[*u*]. In line 5, a priority queue *Q* is initialized to contain all the vertices. Line 7 extracts from queue *Q* the vertex *u* whose key is the minimum. For each vertex *v* adjacent to a vertex *u* (line 8), if the score of the edge (*u*, *v*) is less than the key of *v* (line 9), then edge (*u*, *v*) is added to *MST* (line 10) and the key of *v* is given the score of the edge (*u*, *v*) (line 11).

```
Algorithm CONSTRUCT-MST (NW, S, r)
1.  for each $u \in V[NW]$
2.       do $key[u] \leftarrow \infty$
3.          $\pi[u] \leftarrow$ NIL
4.  $key[r] \leftarrow 0$
5.  $Q \leftarrow V[NW]$
6.  while $Q \neq \emptyset$
7.       do $u \leftarrow EXTRACT-MIN(Q)$
8.          for each $v \in Adj[u]$
9.              do if $v \in Q$ and $S(u,v) < key[v]$
10.                 then $\pi[v] \leftarrow u$
11.                      $key[v] \leftarrow S(u, v)$
12.                      $MST \leftarrow (v, \pi[v])$
```

Fig. 1: Algorithm *CONSTR-MST*

We observe that a vertex *v* is important to a set *S* of vertices in a network, if *S* is existence dependent on *v* through the paths of the MST that connect *v* and *S*. That is, *v* is important to *S*, if the existence of *S* in the network is dependent on the existence of *v*. An existence dependency relation will be added between vertices *u* and *v* if, wherever *v* exists, it is as part of *u*; the presence of *v* implies the presence of *u* [20, 23, 24]. Thus, if the relationship between *u* and *v* represents existence-dependency, *u* and *v* are *closely related*.

A vertex *u* is existence dependent on a vertex *v*, if the removal of *v* causes *u* to be unable to reach *each* other vertex in the network through the paths of MST. If so, the removal of *v* from the MST will require the removal of *u*. Because the removal of *v* from the MST causes *u* to become unable to reach *each* other vertex in the MST, *v* is influential to *u*. If we calculate the sum of the scores of the edges located in the path from *u* to *v* in MST, we find that this sum is the lowest among all other paths located in the paths from *u* to *v*.

ECLfinder identifies for each vertex *v*, the set *S* of vertices, whose existence in MST is dependent on *v*. The removal of *v* causes each vertex *u* ∈ *S* to be unable to reach *each* other vertex through the paths of the MST. Finally, ECLfinder assigns a score to each vertex *v* in the network. The score of the vertex *v* is the number of other vertices, whose existence in MST is dependent on *v*. The score reflects the relative rank/importance of the criminal represented by the vertex *v* in the criminal organization.

B. Identifying the Central Vertices

Vertices are ranked based on their scores described in subsection IV-A. Criminals represented by the top ranked vertices are considered the influential members of the criminal organization. Let the scores of vertices *v* and *u* be *n* and *m* respectively. v controls the flow of information between n criminals, and u controls the flow of information between m criminals. Let $n > m$. Intuitively, v should be ranked higher than u, for the following reasons: (1) v controls the flow of information between more criminals than u, (2) u itself can be existence dependent on v in the MST *(i.e., the criminal represented by v can control the flow of information initiated by the criminal represented by u)*.

Criminal investigators may want to know the immediate leaders of a given list of lower-level criminals in a criminal organization. These criminals are usually the ones in the organization, who carry out crimes; therefore, they are easier to be arrested and incriminated. ECLfinder can also identify the immediate leaders of a given list of lower-level criminals. We use the term "*query vertices*" to refer to a given list of vertices representing lower-level criminals. Let q_1, q_2, … q_n denote a list of query vertices. A criminal represented by a vertex v in a network is considered an immediate leader of the criminals represented by q_1, …, q_n, if: (1) v has the highest score among the vertices located at the convergences of the subtrees of the MST that pass through q_1, …, q_n, and (2) the existence of each of q_1,…, q_n in the MST is dependent on v.

V. Case Studies

We use a partial snapshot of Friendster social network [28] to demonstrate the techniques employed by ECLfinder. The network is publicly available as part of the Stanford Network Analysis Project (SNAP) [28]. Fig. 2 shows the partial snapshot of the network. A vertex in the network represents a user. An edge represents a relationship between two users. The score of an edge is the inverse of the shortest-path betweenness of the edge (recall section III). The bold/thick edges show the path of the Minimum Spanning Tree (MST) of the network.

Fig. 3 shows the same network in Fig. 2 after assigning a score to each vertex in the network using the techniques described in subsection IV-A. The score of a vertex v is the number of other vertices whose existence in MST is dependent on v. The following describe how the scores of some selected vertices in the network in Fig. 3 are determined:

- ➢ The score of vertex STEVEN is 4, because the following four vertices are existence dependent on STEVEN in MST: THOMAS, JOHN, LARRY, and JERRY. That is, the removal of vertex STEVEN will cause the four vertices to be unable to reach each of the remaining vertices connected with the root vertex through the paths of the MST. Observe that vertex JASON, for example, is unaffected by the removal of vertex STEVEN, since it can still reach each other vertex connected with the root vertex through the paths of the MST.
- ➢ The score of vertex PETER is 9. This is because the removal of vertex PETER will cause the following 9 vertices to be unable to reach each of the remaining vertices connected with the root vertex through the paths of the MST: ERIC, JEFF, SCOTT, JASON, STEVEN, JOHN, THOMAS, LARRY, and JERRY.

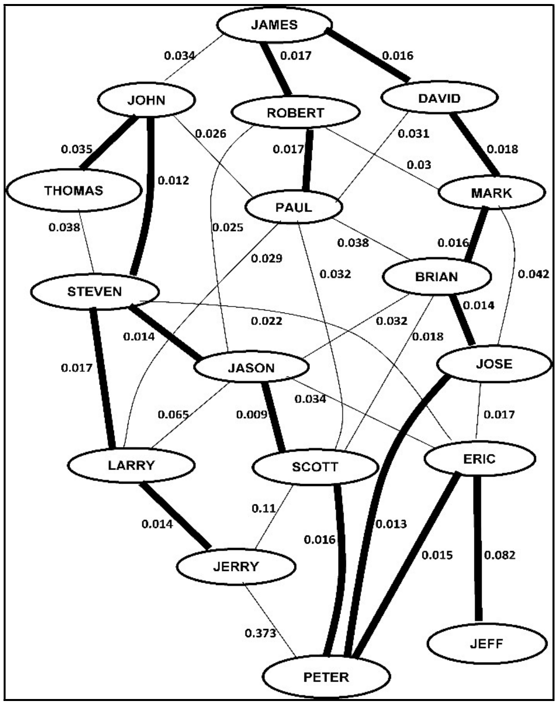


Fig. 2: A partial snapshot of Friendster social network [28]. The score of an edge is the inverse of the shortest-path betweenness of the edge. The bold/thick edges are the paths of the MST of the network

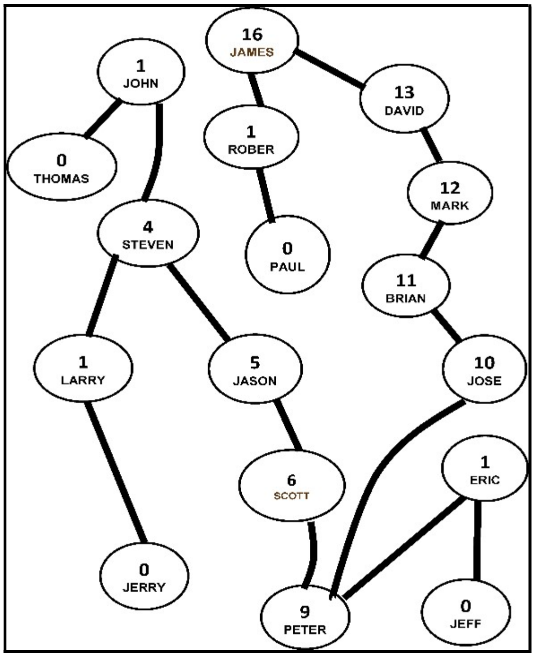


Fig. 3: The partial social network presented in Fig. 2 after assigning a score to each vertex. The number inside each vertex represents the vertex's score. The score of a vertex v is the number of other vertices whose existence in MST is dependent on v. Edges in the figure represent the paths of the MST

Table 1 shows the names of the 17 users represented by the 17 vertices in the partial snapshot of the Friendster social network shown in Fig. 2. The names in the table are ranked based on the scores of the vertices representing them and shown in Fig. 3 (see vertices' scores in Fig. 3). The top ranked users in the table are the influential ones in the social network.

TABLE 1
THE 17 USERS REPRESENTED BY THE 17 VERTICES IN THE PARTIAL NETWORK SHOWN IN FIG. 2 RANKED BASED ON THEIR SCORES SHOWN IN FIG. 3

Rank	Score	Criminal Name
1	16	JAMES
2	13	DAVID
3	12	MARK
4	11	BRIAN
5	10	JOSE
6	9	PETER
7	6	SCOTT
8	5	JASON
9	4	STEVEN
10	1	JOHN, ROBERT, LARRY, ERIC
14	0	THOMAS, PAUL, JERRY, JEFF

Consider Fig. 3 and the following query: *Q*("THOMAS", "LARRY"). The query asks for the immediate leader of THOMAS and LARRY. As Fig. 4 shows, STEVEN is the immediate leader, because of the following: (1) vertex STEVEN is located at the convergence of the subtrees of the MST that passes through vertices THOMAS and LARRY *(recall the last paragraph in subsection IV-B)*, and (2) the existence of vertices THOMAS and LARRY in the MST is dependent on vertex STEVEN *(the removal of vertex STEVEN will cause the two vertices to be unable to reach each of the vertices in the other subtree containing the root vertex)*.

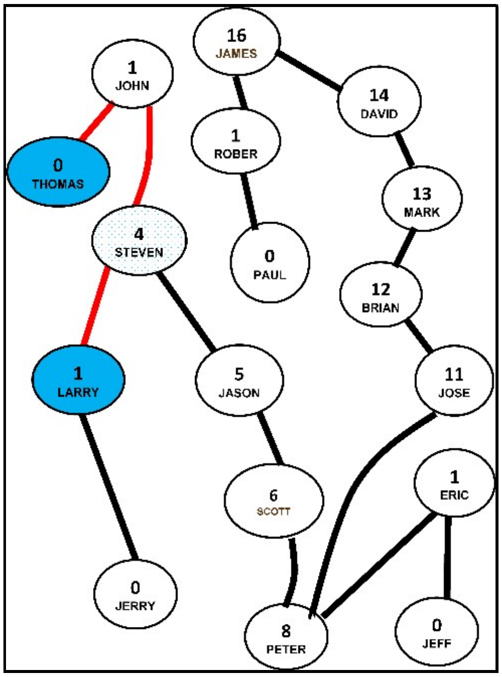


Fig. 4: The red paths show that vertex STEVEN is located at the convergence of the subtree of the MST that passes through vertices THOMAS and LARRY

Consider Fig. 3 and the query: *Q*("JERRY", "ERIC"). The query asks for the immediate leader of JERRY and ERIC. As Fig. 5 shows, PETER is the immediate leader, because of the following: (1) vertex PETER is located at the convergence of the subtrees of the MST that passes through vertices JERRY and ERIC *(recall the last paragraph in subsection IV-B)*, and (2) the existence of vertices JERRY and ERIC in the MST is dependent on the existence of vertex PETER.

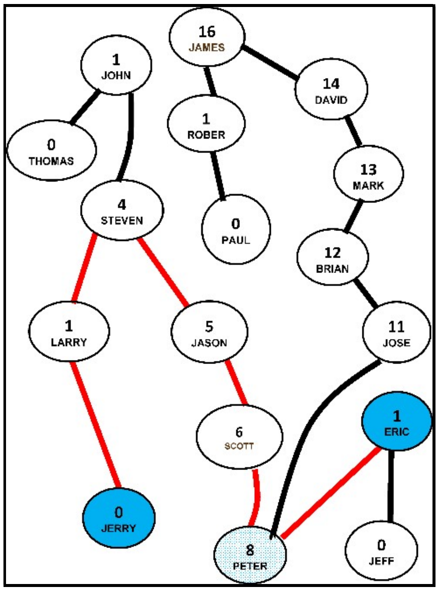


Fig. 5: The red paths show that vertex PETER is located at the convergence of the subtree of the MST that passes through vertices JERRY and ERIC.

VI. EXPERIMENTAL RESULTS

We implemented ECLfinder in Java, run on Intel(R) Core(TM) i7 processor, with a CPU of 2.70 GHz and 16 GB of RAM, under Windows 10. We evaluated the quality of ECLfinder by comparing it with LogAnalysis [8], CrimeNet Explorer [12], and our previously proposed system SIIMCO [19]. The following are brief descriptions of the three systems:

- LogAnalysis [8]: It employs Girvan & Newman [11] algorithms to identify the degree of relationships between vertices representing criminals in a criminal network. It can identify the influential members in criminal organization. It can use mobile phone communication data that belongs to a criminal organization to construct a network depicting the relationships between the criminals in the organization.
- CrimeNet Explorer [12]: It uses hierarchical clustering techniques to partition a network representing a criminal organization into subnetworks based on the strength of the relationships between the vertices in each subnetwork. It employs the Closeness, Degree, and Betweenness centrality metrics to determine the important vertices in a subnetwork. It first identifies the degree of relationship between vertices using the shortest path algorithm and Blockmodeling [3].
- SIIMCO [19]*:* It uses a formula that quantifies the degree of influence of each criminal in a criminal organization relative to all other criminals. Given a set of query vertices, SIIMCO determines the *relative importance* of each vertex in the network with respect to the query vertices, using formulas that quantify the degree of influences of a vertex. One of the key differences between ECLfinder and SIIMCO is that SIIMCO adopts vertex-centric approach while ECLfinder adopts edge-centric approach. In SIIMCO, the importance of a vertex *v* is determined based on the importance of the vertices connecting *v* with the network. In ECLfinder, the importance of a vertex *v* is determined based on the importance of the edges connecting *v* with the network, using the concept of *existence dependency*.

A. Compiling Datasets for the Evaluation

We used the following two real-world communication datasets: Krebs's 9/11 dataset [26, 27] and Enron email corpus [9]. We converted the datasets into networks. Below are brief descriptions of the datasets:

- *Krebs's 9/11 dataset [26, 27]:* We used the Krebs's well-known dataset of the 9/11 incident. The 9/11 were a series of four coordinated terrorist attacks on the United States on the morning of September 11, 2001. We used a weighted version of the Krebs' 9/11 network dataset [26, 27]. The network consists of 62 nodes representing all individuals involved in the incident. The network contains 153 edges. These edges represent reported interactions between the actors involved in the incident. The average node degree in the network is 4.9. The weight of an edge reflects the degree of communications between the two individuals represented by the two nodes connected by the edge.
- *Enron email dataset [9]:* Enron email corpus surfaced following a criminal scandal involved top Enron employees. The corpus includes 619,446 email messages belonging to 158 Enron employees. We cleaned the dataset by removing emails that were exchanged between people other than the 158 employees. The remaining dataset includes 200,136 emails from 151 Enron employees.

B. Evaluating the Accuracy of Identifying the Influential Members of a Criminal Organization

1) Calculating the Recall, Precision, and F-value of the Systems by Comparing their Results with Results Determined by the Standard Network Metrics

In this test, we measure the performance of the systems by comparing their results with the results determined by the standard Closeness, Betweenness, In Degree, and Out Degree Centrality metrics. Degree is the number of ties that a vertex has. Vertices with high degree centralities are central in the network. The betweenness centrality of a vertex *v* is the number of shortest paths between other vertices that pass through *v*. Closeness centrality is the length of the shortest path to all other vertices. It measures how a vertex is close to other vertices. We calculated the Recall, Precision, and F-value using the following standard metrics:

$$\text{Recall}=\frac{N_s^c}{N_m^{top}},\qquad \text{Precision}=\frac{N_s^c}{N_s^{top}},\qquad F-value=\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}$$

where N_s^c is the number of *correct* vertices returned by a system, N_m^{top} is the number of *actual correct* vertices, and N_s^{top} is the number of vertices returned by a system. Let L_{top} be the list of top vertices returned by a standard network metric and let L_s be the list of correct vertices returned by a system. $N_s^c \subseteq L_{top}$ and $N_m^{top} = |L_{top}|$.

We submitted the networks representing the Krebs's 9/11 and Enron datasets to the four standard network metrics, and we also submitted the same networks to each of the four systems. We then calculated the Recall, Precision, and F-value of the results returned by each of the four systems. The results are shown in Tables II and III.

TABLE II
PERFORMANCE OF THE SYSTEMS USING THE 9/11 DATASET COMPUTED BASED ON THE TOP VERTICES RETURNED BY THE STANDARD NETWORK METRICS

		Recall	Precision	F-value
ECLfinder	Closeness Centrality	0.66	0.61	0.63
SIIMCO		0.62	0.55	0.58
CrimeNet Explorer		0.54	0.58	0.56
LogAnalysis		0.51	0.49	0.50
ECLfinder	Betweenness Centrality	0.59	0.57	0.58
SIIMCO		0.55	0.50	0.52
CrimeNet Explorer		0.49	0.43	0.46
LogAnalysis		0.39	0.43	0.41
ECLfinder	In Degree Centrality	0.66	0.68	0.67
SIIMCO		0.64	0.59	0.61
CrimeNet Explorer		0.52	0.46	0.49
LogAnalysis		0.52	0.54	0.53
ECLfinder	Out Degree Centrality	0.71	0.67	0.69
SIIMCO		0.69	0.55	0.61
CrimeNet Explorer		0.57	0.51	0.54
LogAnalysis		0.66	0.61	0.63

TABLE III
PERFORMANCE OF THE SYSTEMS USING ENRON DATASET COMPUTED BASED ON THE TOP VERTICES RETURNED BY THE STANDARD NETWORK METRICS

		Recall	Precision	F-value
ECLfinder	Closeness Centrality	0.58	0.50	0.54
SIIMCO		0.52	0.46	0.49
CrimeNet Explorer		0.37	0.30	0.33
LogAnalysis		0.40	0.34	0.37
ECLfinder	Betweenness Centrality	0.44	0.37	0.40
SIIMCO		0.46	0.39	0.42
CrimeNet Explorer		0.34	0.26	0.29
LogAnalysis		0.44	0.39	0.41
ECLfinder	In Degree Centrality	0.69	0.67	0.68
SIIMCO		0.64	0.61	0.62
CrimeNet Explorer		0.40	0.34	0.37
LogAnalysis		0.58	0.56	0.57
ECLfinder	Out Degree Centrality	0.65	0.59	0.62
SIIMCO		0.61	0.52	0.56
CrimeNet Explorer		0.49	0.44	0.46
LogAnalysis		0.45	0.38	0.41

2) Calculating the Euclidean Distances between the Results of each System and the Results of the Network Metrics

We measured the average Euclidean Distance between the top ranked *n* vertices returned by a system and the corresponding top ranked *n* vertices returned by a standard network metric. We considered *n* equals 5, 10, and 15. We used the following Euclidean distance measure.

$$d(\sigma_m,\sigma_s)=\sum_{x\in N_m^{top}}|\,\sigma_m(v)-\sigma_s(v)\,|$$

- N_m^{top} are the top n vertices returned by network metric *m*.
- $\sigma_m \in [0,1]^{|N_m^{top}|}$ and $\sigma_s \in [0,1]^{|N_m^{top}|}$ are the top *ranked n* vertices returned by metric *m* and system *s*, respectively.
- $\sigma_m(v)$, $\sigma_s(v)$ are the position of vertex $v \in N_m^{top}$ in the lists σ_m and σ_s respectively. Fig. 6 shows the average Euclidean Distances using the Krebs's 9/11 and Enron datasets.

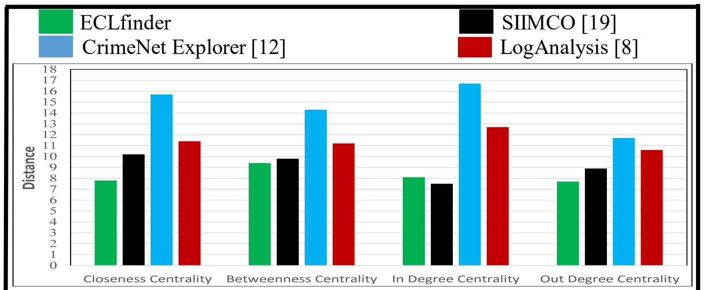
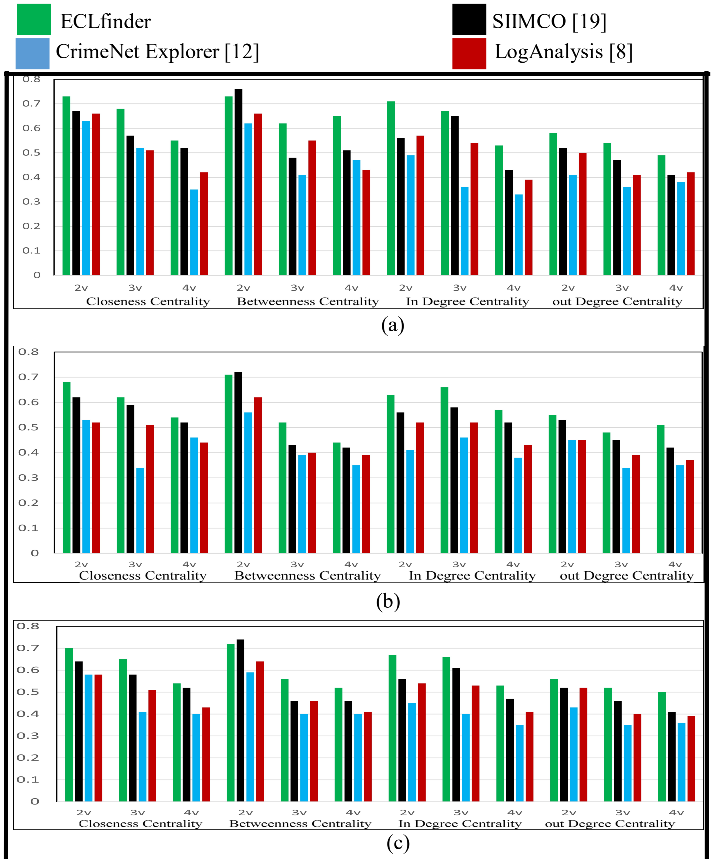


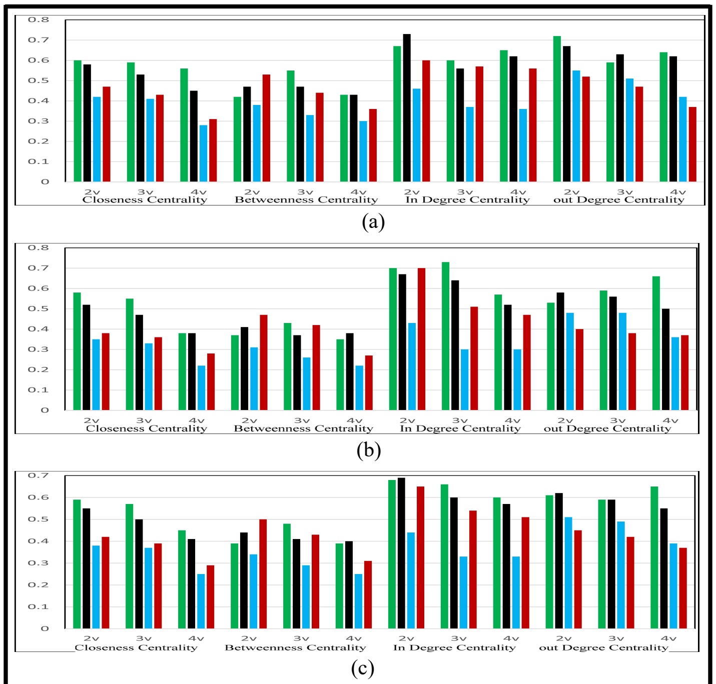
Fig. 6: Average Euclidean distances between the results returned by each of the four systems and the results returned by the standard network metrics using Krebs's 9/11 and Enron datasets.

C. Evaluating the Accuracy of Identifying the Immediate Leaders of Lower Level Criminals in a Criminal Organization

We evaluated the accuracy of the four systems for identifying the most important vertices to a given list of vertices in the networks representing the Krebs's 9/11 and Enron datasets. We randomly selected 50 lists of 2-query vertices, 50 lists of 3-query vertices, and 50 lists of 4-query vertices from each of the two networks. We submitted the query vertices and the networks representing the datasets to the standard network metrics and to the four systems. We considered only the top 5 vertices returned by each of the metrics as the list N_s^{top} (recall section VI-B-1). We compared the top 5 vertices returned by each system with the list l_{top}. We then calculated the Recall, Precision, and F-value of each system. Figs. 7 and 8 show the results for the Krebs's 9/11 and Enron datasets, respectively.



Figs. 7: (a) Recall, (b) Precision, and (c) F-value of the four systems for identifying the important vertices to a given list of query vertices using the Krebs's 9/11 dataset. In the figure, 2v, 3v, and 4v denote the following: 2 query vertices, 3 query vertices, and 4 query vertices respectively.



Figs. 8: (a) Recall, (b) Precision, and (c) F-value of the four systems for identifying the important vertices to a given list of query vertices using the Enron dataset. In the figure, 2v, 3v, and 4v denote the following: 2 query vertices, 3 query vertices, and 4 query vertices respectively.

D) Discussion of the Results

The following are our observations of the experimental results using the Krebs's 9/11 dataset:

a) ECLfinder was able to identify the key nodes in the network not only because they have more connections, but also because their links to other nodes in the network are much stronger compared to the links of the less central nodes.

b) ECLfinder was able to identify the nodes in the network representing the following most influential (i.e., central) actors in the incident: Atta, Al-Shehi, Jarrah, Khemais, Moussaoui, Hanjour, Al-Hazmi, Al-Shibh, and Essabar.

c) ECLfinder was able to identify the node representing Atta, the ringleader of the hijackers, as the most central node in the network.

d) Each of the top four nodes identified by ECLfinder represents one of the hijackers on one of the four planes.

e) ECLfinder ranked the nodes representing Khemais, Moussaoui, and Jarrah very high. It has been revealed that Khemais and Moussaoui served as coordinators between the hijackers and other actors involved in the incident. It has also been revealed that Jarrah was one of the masterminds of the 9/11 plot

The top five nodes returned by ECLfinder in the Enron network represent the following actors in the Enron scandal:

- Arthur Andersen (auditor).
- Kenneth Lay (CEO).
- Sheila Kahanek (accountant).
- Andrew Fastow (financial officer).
- Jeffrey Skilling (COO).

Three of these five individuals have been charged and found guilty of various conspiracy and accounting frauds.

As Figs. 6-8 and Tables II and III show, ECLfinder outperformed the other three systems. Based on our observations of the experimental results, we attribute the performance of ECLfinder over the three systems to the following limitations of LogAnalysis, CrimeNet Explorer, and SIIMCO:

1. LogAnalysis limitations:

a) It does not work well for clustering large-size networks. The results showed that it clusters small-size networks more accurately than large-size ones.

b) It is biased to globular clusters.

c) It cannot detect and undo incorrect clustering that was done at an early stage.

d) If clusters have different sizes, it may not work well.

e) Due to the nature of its techniques, some vertices may not contribute to the overall importance value of a vertex (Incomplete Contribution) and some vertices may contribute unequally to the overall importance value of a vertex (Inconsistent Contribution).

2) CrimeNet Explorer limitations:

Let (*u*, *v*) be the most important incoming edge to vertex *v*. CrimeNet Explorer determines the weight of vertex *v* based *solely* on the weights of edge (*u*, *v*) and vertex *u*.

3) SIIMCO limitations:

It does not work well when the network consists of a large number of vertices and edges. We reached this conclusion after comparing SIIMCO with ECLfinder using datasets with various number of vertices and edges. We used for the comparison the following three real-world datasets compiled by the Stanford Network Analysis Project (SNAP) [28]: *com-Friendster* (65,608,366 nodes, 1,806,067,135 edges), *com-Orkut* (3,072,441 nodes, 117,185,083 edges), and *com-Amazon* (334,863 nodes, 925,872 edges). We measured the performance of SIIMCO and ECLfinder by comparing their results with the results returned by the standard Closeness, Betweenness, In Degree, and Out Degree Centrality metrics using the same procedure described in subsection VI-B-1. We observed that ECLfinder achieved the highest performance over SIIMCO when the *com-Friendster* dataset was used. ECLfinder achieved the second highest performance over SIIMCO when the *com-Orkut* dataset was used. The least performance of ECLfinder over SIIMCO was when the *com-Amazon* dataset was used.

VII. Conclusion

We introduced in this paper a forensic analysis system called ECLfinder. The system can determine the influential members of a criminal organization as well as the immediate leaders of a given list of lower-level criminals associated with the organization. First, ECLfinder constructs a network representing a criminal organization from either MCD that belongs to the organization or from crime incident reports containing information about the organization. A vertex in such a network represents an individual criminal and an edge represents the relationship between two criminals. ECLfinder identifies the influential members of the criminal organization by determining the important vertices in the network representing the organization, using the concept of existence dependency. A vertex *v* is influential to a set *S* of vertices in the network, if the existence of *S* in the network is dependent on the existence of *v* through the paths of the MST that connect *v* with *S*. Each vertex *v* is assigned a score, which is the number of vertices in set *S*. Vertices are ranked based on their scores. Criminals represented by the top ranked vertices are considered the influential members of the criminal organization. We experimentally compared ECLfinder with SIIMCO [19[, CrimeNet Explorer [12], and LogAnalysis [8] for identifying the important vertices in networks. Results revealed that ECLfinder outperforms the three systems.

References

[1] Agreste, S., Catanese, S., De Meo, P., Ferrara, E., & Fiumara, G. (2015). Network Structure and Resilience of Mafia Syndicates. arXiv preprint arXiv:1509.01608.

[2] BREIGER, R. L. 2004. The analysis of social networks. In *Handbook of Data Analysis*, M. A. Hardy and A. Bryman, Eds. Sage Publications, London, U.K. 505–526.

[3] BREIGER, R. L., BOORMAN, S. A., AND ARABIE, P. 1975. An algorithm for clustering relational data, with applications to social network analysis and comparison with multidimensional scaling. *J. Math. Psych. 12*, 328–383.

[4] BAKER, W. E. AND FAULKNER R. R. 1993. The social organization of conspiracy: Illegal networks in the heavy electrical equipment industry. *Amer. Soc. Rev. 58*, 837–860.

[5] CHEN, H. AND LYNCH, K. J. 1992. Automatic construction of networks of concepts characterizing document databases. *IEEE Trans. Syst. Man Cybernet. 22*, 885–902.

[6] CHEN, H., ZENG, D., ATABAKHSH, H., WYZGA, W., AND SCHROEDER, J. 2003. Coplink: Managing law enforcement data and knowledge. *Commun. ACM 46*, 28–34.

[7] Catanese, S., Ferrara, E., & Fiumara, G. (2013). Forensic analysis of phone call networks. Social Network Analysis and Mining, 3(1), 15-33.

[8] E. Ferrara, P. De Meo, S. Catanese, and G. Fiumara, “Detecting criminal organizations in mobile phone networks,” *Expert Systems with Applications*, vol. 41, no. 13, pp. 5733–5750, 2014.

[9] Enron Email Dataset. Available at: http://www-2.cs.cmu.edu/~enron/.

[10] Ferrara, E., Catanese, S., & Fiumara, G. (2015). Uncovering Criminal Behavior with Computational Tools. In Social Phenomena (pp. 177-207). Springer International Publishing.

[11] Girvan, M., Newman, M. (2002). Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12), 7821.

[12] J. J. Xu and H. Chen, “CrimeNet explorer: A framework for criminal network knowledge discovery,” *ACM Trans. Inf. Syst.*, vol. 23, no. 2, pp. 201–226, Apr. 2005.

[13] J. Pattillo, N. Youssef, and S. Butenko, “Clique relaxation models in social network analysis,” in *Handbook of Optimization in Complex Networks*. Springer, 2012, pp. 143–162.

[14] L. Langohr, “Methods for finding interesting vertices in weighted graphs,” Ph.D. dissertation, 2014.

[15] MCANDREW, D. 1999. The structural analysis of criminal networks. In *The Social Psychology of Crime: Groups, Teams, and Networks*. D. Canter and L. Alison, Eds. Dartmouth Publishing, UK, 53–94.

[16] Memon, Bisharat, *Identifying Important Nodes in Weighted Covert Networks Using Generalized Centrality Measures*. 2012 European Intelligence and Security Informatics Conference (EISIC 2012).

[17] Stanford Tokenizer, Part-of-Speech Tagger, and Named Entity Recognizer. Downloaded from: http://nlp.stanford.edu/software/

[18] Shang, X., Yuan, Y. *Social Network Analysis in Multiple Social Networks Data for Criminal Group Discovery*. 2012 Conference on

Cyber-Enabled Distributed Computing and Knowledge Discovery.

[19] Taha, K., and Yoo, P. “SIIMCO: A Forensic Investigation Tool for Identifying the Influential Members of a Criminal Organization”. *IEEE Transactions on Information Forensics & Security*, 2015, Vol. 11, issue 4, pp. 811 - 822.

[20] Taha, K. "Determining the Semantic Similarities among Gene Ontology Terms". *IEEE Journal of Biomedical and Health Informatics (IEEE J-BHI)*, 2013, Vol. 17, Issue 3, pp. 512 - 525.

[21] Taha, K. and Elmasri, R. "BusSEngine: A Business Search Engine." *Knowledge and Information Systems: An International Journal (KAIS)*, 2010, LNCS, Springer, Vol. 23, No. 2, pp. 153-197.

[22] Taha, K. and Elmasri, R. "CXLEngine: A Comprehensive XML Loosely Structured Search Engine." In proceedings of *Database technologies for handling XML information on the web (DataX'08), France, March 2008.*

[23] Taha, K., Homouz, D., Al Muhairi, H., and Al Mahmoud, Z. "GRank: A Middleware Search Engine for Ranking Genes by [502] Relevance to Given Genes". *BMC Bioinformatics* 2013, 14:251.

[24] Taha, K. and Elmasri, R. "SPGProfile: Speak Group Profile." *Information Systems (IS)*, 2010, Elsevier, Vol. 35, No. 7, pp. 774-790.

[25] U. K. Wiil, J. Gniadek, N. Memon; *Measuring Link Importance in Terrorist Networks. Social Network Analysis*, Conference On Advances in Social Networks Analysis and Mining, ASONAM 2010.

[26] V. E. Krebs, “Uncloaking terrorist networks,” *First Monday*, vol. 7, pp. 4–11, 2002.

[27] V. E. Krebs, “Mapping networks of terrorist cells,” *Connections*, vol. 24 (3), pp. 43–52, 2002

[28] Web Archive Project. Stanford Large Network Dataset Collection (online). Available at: http://snap.stanford.edu/data/com-Friendster.html

Kamal Taha is an Assistant Professor in the Department of Electrical and Computer Engineering at Khalifa University, UAE, since 2010. He received his Ph.D. in Computer Science from the University of Texas at Arlington, USA, in March 2010. He has over 70 refereed publications that have appeared in prestigious top ranked journals, conference proceedings, and book chapters. Fifteen of his publications have appeared (or are forthcoming) in IEEE Transactions journals. He was as an Instructor of Computer Science at the University of Texas at Arlington, USA, from August 2008 to August 2010. He worked as Engineering Specialist for Seagate Technology, USA, from 1996 to 2005 *(Seagate is a leading computer disc drive manufacturer in the US)*. His research interests span Information Forensics & Security, bioinformatics, information retrieval, data mining, and databases, with an emphasis on making data retrieval and exploration in emerging applications more effective, efficient, and robust. He serves as a member of the Program Committee, editorial board, and review panel for a number of international conferences and journals, some of which are IEEE and ACM journals. He is a Senior Member of IEEE.

Paul D. Yoo received his PhD in Engineering and IT from the University of Sydney (USyd) in 2008. He was a Research Fellow in the Centre for Distributed and High Performance Computing, at USyd from 2008 to 2009, and PHD Researcher (Quantitative Analysis) at the Capital Markets CRC, administered by the Australia Federal Dept. for Education, Science and Training, from 2004 to 2008. He was with the ATIC-Khalifa Semiconductor Research Center, KUSTAR from 2009 to 2014. From 2014 to 2016 he worked as a Lecturer at the Data Science Institute, Bournemouth University, U.K. He is currently a Lecturer at Cranfield University and Defence Academy, U.K. He holds over 60 prestigious journal and conference publications and is currently actively involved in editorial board, technical program committees, and review panels of the data science and analytics areas for top conference and journal.