How to beat terrorism efficiently: identification of set of key players in terrorist networks

Marco Pietro Abrate, Natalie Bolón Brun, Shahow Kakavandy, Jangwon Park
École Polytechnique Fédérale de Lausanne, 2019

I. INTRODUCTION

Proliferation of terrorism in recent years has led people to believe it as a real threat to their livelihood. Vital to the success of such terrorist organizations are the cohesiveness and ability to communicate efficiently within their respective terrorist networks. To make these networks vulnerable, identifying sources of such properties is an imperative mission and hence becomes the focus of this report. More technically, we seek to develop an appropriate methodology to evaluate the importance of each terrorist to the effectiveness of the network as a whole, and identify an optimal set of key terrorists that one should target in order to debilitate it.

II. PROBLEM STATEMENT

The project aims to find points of vulnerability in the terrorist relations network that one can exploit to reduce its overall effectiveness. We define vulnerability in the sense of key terrorists in the network whose absence will fragment it as much as possible. Similarly, we define it also in the sense of key terrorists who, if fed with deliberate misinformation, are best positioned to spread it most quickly and widely. As such, we employ a key player approach which is broken into two separate problems:

1) **Fragmentation**: identify a set of key terrorists that best fragments the network when removed.
2) **Information flow**: identify a set of key terrorists who are best positioned in the network to spread false information most efficiently.

III. DATA PROCESSING & CLEANING

The original dataset is acquired from LINQS [1]. The network it provides encodes relations between terrorists. Its nodes represent relations (labeled as colleague, congregate, contact or family) while the edges represent names of terrorists. Given the nature of the edges, they are unweighted and undirected. The original network has 851 nodes and 8,592 edges and its largest connected component consists of 665 nodes and 6,552 edges.

The interpretation of this first representation suggested by the original analysis of the network [2] is arguably counter-intuitive. Therefore, the first data processing step is to invert the network such that the nodes represent the terrorists, between whom is an edge only if they are related in some way. However, disconnected components in the original network must be inverted and analyzed separately. In this project, we invert and analyze the largest component which is sufficiently large to represent all the major characteristics of the entire network (and hereby refer to the largest component of the original network simply as the 'original network'). The following describes the network inversion process:

1) Extract terrorist names from the unique ID of each node in the largest component of the original network.
2) Initialize an adjacency matrix whose size is equal to the number of unique terrorists found in step 1.
3) Set $a_{ij} = 1$ between terrorists i and j if they belonged to the same node in the original network.

The unique ID of each node in the original network is a URL with terrorist names embedded in it. We extract the names by parsing the URL on special characters (e.g. #). An important finding in this process is that each unique ID will contain at most two terrorist names. This implies that no matter how high the degree of the node in the original network is, all its edges represent at most two unique terrorists. When the unique ID is missing a terrorist name, we discovered that it always has a unique datetime string in its place. Therefore, we tentatively use the datetime string as the name for that particular terrorist.

The inverted network has a size of 244 nodes and 661 edges, a significant reduction from the original network. This verifies that many edges in the original network are duplicate terrorists. Additionally, 126 of the 244 unique terrorists did not have known names and were thus replaced with unique datetime strings.

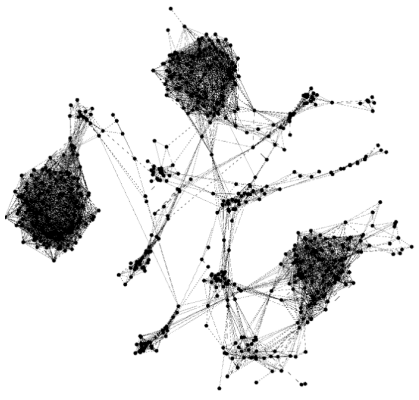

Fig. 1: Largest component of the original network