Social Networks 51 (2017) 178–188

Contents lists available at ScienceDirect

Social Networks

journal homepage: www.elsevier.com/locate/socnet

# A graph database framework for covert network analysis: An application to the Islamic State network in Europe

Alexander Gutfraind [a,*], Michael Genkin [b]

[a] University of Illinois at Chicago, Loyola University Medical Center, Uptake Technologies, Inc., United States
[b] School of Social Sciences, Singapore Management University, Singapore

ARTICLE INFO

*Article history:*
Available online 25 November 2016

*Keywords:*
Covert network
Terrorist network
Graph database
Terrorism
Islamic State

ABSTRACT

This paper proposes a new framework, based on graph database theory, for encoding complex data on covert networks, mapping their structure, and conducting a sensitivity analysis. The framework is then applied to reconstruct the terrorist network of the 2015–2016 attacks in Paris and Brussels, and related plots in Europe by the Islamic State group. The resulting network was found to be qualitatively different from the ideologically-related Al-Qaeda network, having a lower secrecy and a lower mean degree, under different network-generating assumptions.



## 1. Introduction

How could a covert social network be measured with reasonable confidence? Unlike data collection for overt social networks, it is rarely possible, in covert networks, to collect respondent-generated ties via name generators and name interpreters or even to measure ties in a uniform manner. Instead researchers are forced to infer and code their network structure from incomplete, indirect, and uncertain information (Gerdes, 2015). Multiple coding decisions need to be made regarding how the ties between nodes should be measured in addition to which nodes to include. This introduces subjectivity and affects reproducibility. To some extent this is the bane of all network studies, but covert networks pose a special challenge given the secondary nature of the data.

To address this problem, the paper proposes and demonstrates a framework to encoding and analyzing covert networks, which has the advantage of systematically accounting for the uncertainty and ambiguity of covert network data. The framework relies on newly-developed methods from graph database theory (see Angles and Gutierrez 2008; Robinson et al., 2015). The approach allows the researcher to encode raw data in multi-modal form and then use the powerful tools of graph databases to project the data into a social network. Once the network has been projected from the graph database, it is analyzed using existing methods of social network analysis. The framework allows the user to conduct a sensitivity analysis to examine whether the network structure depends upon the way the researcher chose to generate their network in terms of how ties were recorded, how nodes were counted, and how sources for the information were weighted.

This paper is organized as follows. First, we begin by describing the problem of using covert network data. Next, we introduce the graph database as a tool for studying covert social networks and compare it to more familiar methods. Third, we introduce sensitivity analysis based on the graph database framework. Fourth, we apply this method to the Islamic State (IS) network in Europe that was responsible for the November 2015 Paris Attack and the March 2016 Brussels Attack and compare it to other covert networks. Finally, we conclude by discussing the findings on characterizing this network, the use of graph databases for covert network analysis, as well as the broader implications of graph databases for developing social network data standards and its potential for big data analytics.

### 1.1. The Challenges of Covert Network Data

Social network analysis faces a variety of unique data challenges, especially when it comes to sociocentric data (see Borgatti et al., 2013; Robins 2015).These range from specifying boundaries (Laumann et al., 1989), to adequate sample size (Costenbader and Valente, 2003; Frank, 2011) to issues with various forms of missing data (Kossinets, 2006). The problems are amplified with covert networks because the actors are consciously attempting to conceal or falsify their identities, their attributes, or their connections (Gerdes, 2015). As a result, the instruments of collecting network data directly from respondents such as name generators or name interpreters are usually not available. Data on covert networks typically come from secondary sources. In the case of terrorist or

* Corresponding author at: University of Illinois at Chicago, 1603 W. Taylor St, Chicago, IL 60612, United States.
*E-mail address:* agutfraind.research@gmail.com (A. Gutfraind).

http://dx.doi.org/10.1016/j.socnet.2016.10.004
0378-8733/

organized crime networks the data are inferred retroactively from sources such as physical and electronic surveillance documents (Morselli, 2009); court records (Baker and Faulkner, 1993); or news reports (Krebs, 2002). As with any data, the analyst is forced to make coding decisions before social network analysis can even begin. There are two issues that arise, which should be kept separate. The first is at the level of analysis and the second is at the level of measurement.

First, there is the issue of what counts as a tie and to what extent should different types of ties be distinguished from one another. Some researchers have criticized the tendency in covert network analysis to amalgamate multiplex ties such as kinship, friendship, and organizational roles into a uniplex relation (see Gerdes 2015, chapter 2). For example, if actor A has a family tie to actor B as well as an operational tie with him, the combined uniplex relation is often given a double score. Doing so involves throwing out information and assigning weights arbitrarily, thus introducing the analyst's own biases (ibid). Lumping diverse relationships between actors as equivalent "ties" without theoretical justification introduces bias at the level of coding analysis.

The second issue is whether the nodes included and their relationships are in fact correctly measured. There is always the problem of missing data and whether certain nodes or relations are simply not known – the false negatives. But there is also the problem of how accurate are the relationships that are "known" – the false positives. Measurement error is especially pronounced for covert networks due to the secondary nature of the data collection. This paper introduces a framework that seeks to reduce measurement error, though it address both measurement and analysis errors.

For the purposes of this paper we refer to the problem of measuring and representing relationships in a covert network as the problem of covert network forensics. Much like a detective trying to piece together the details of a crime while sifting through a multitude of clues, covert networks forensics involves the systematic piecing together of information about relationships between nodes. Data about a covert network are usually obtained after the network has already carried out its mission. In the case of assembling information from disparate sources, there is a great deal of uncertainty as to which ties and nodes to include. Thus, what is needed is a framework that has the following four properties.

*Documentation*

The ability to link specific network elements (nodes or ties) to the raw source data as metadata. In this way the data are precisely documented and can be reproduced and easily re-examined.

*Complex representation*

The ability to store diverse entities and relations from multiple modes along with data on multiple types of edges. This is useful as the available sources often describe multi-modal relations.[1]

*Reproducibility*

The ability to efficiently reproduce coding decisions by other researchers. Ideally all the coding decisions are documented in a command script file.

*Multiple projection*

The ability to use raw data in multiple ways when deriving the network's nodes and edges. This allows the analyst to rerun her analysis and to test the robustness of one's network to different network-generating assumptions. This is especially important for covert networks because there are many arbitrary decisions that are made regarding tie measurement.

Traditional packages for social network analysis are not well-suited for these tasks because it is cumbersome to store so many different kinds of data: multiple edge attributes, multiple modes, and non-network data. Indeed many covert network datasets that are publicly available, even those derived from open sources, are poorly documented and very difficult to reproduce.

### 1.2. Graph databases

To respond to these challenges, this paper uses the theory of graph databases, a relatively new methodology from computer science, to perform covert network forensics. Generally, a graph database is a knowledge representation system which codes knowledge using nodes and edges, rather than storing tables of rows and columns as found in conventional databases (Angles and Gutierrez 2008; Robinson et al., 2015). Graph databases have recently matured as a technology and are capable of exceeding, in some respects, methods of conventional databases, even when working with data such as tables, text, and images. Because the internal structure of graph databases is in the form of a network, they are very well-suited to represent the information about covert networks, including members, activities, events and the relationships among them, as well as the attributes of the entities and relationships (Fig. 1). Graph databases can also contain meta-information *about* the nodes and edges, including the full text, image or movie containing the evidence, and this information can be considered when reconstructing the covert network.

Software implementations of graph databases are also equipped with a powerful and versatile query language for extracting information, analogous to how relational databases are queried using the SQL language. The query language of the graph database allows systematic extraction of information from the available raw data – a method not possible using social network analysis packages. Using a query, it is possible to determine the members of a covert network, and the relationships between them by listing only members and relationships that satisfy a particular condition. Furthermore, as new data are added to the database, the query can instantly (i.e., with one command) report the updated structure of the covert network based on any new information. The query also facilitates easy sensitivity analysis to determine whether the covert network is dependent upon the way the data are being coded (Kossinets, 2006). The social network resulting from the database query is then visualized and analyzed using existing social network tools and methods.

While the most advanced currently available network software packages have the ability to store multiple types of nodes and edges as well as their attributes, it is not generally possible in these systems to perform automated extraction of network information based on complex logical criteria. By contrast, the graph database is well-suited to store and query detailed temporal, spatial and other attributes about the nodes and relations. For example, a query can list all pairs of persons who were present in the same location within 7 days of a terrorist attack and who also visited a specific country in the past 5 years. In general, a graph database query returns any logical path (a set of one or more linked

[1] For example, suppose node A and node B shared an apartment in Molenbeek while node C and node D lived in the same area of Paris. If both pairs are coded as "co-presence", one throws away the data that indicates co-presence with respect to specific locations (e.g. Molenbeek, Paris). This information might be relevant for subsequent analysis or by other researchers. While it is possible for the analyst to go back to her raw data and re-code or for other researchers to contact the original coder for their raw data, it is not efficient to do so. Ideally, the analyst should input as much as possible of the relevant raw data into machine-readable format while throwing away the least possible amount of information, at the stage of data coding.