# Poster: Disentangling Cuts on Surveillance Graphs

William Schulze
*Computer Science and Engineering*
*Michigan State University*
*East Lansing, MI, USA*
*Email: schulzew@msu.edu*

Sandeep Kulkarni
*Computer Science and Engineering*
*Michigan State University*
*East Lansing, MI, USA*
*email: sandeep@msu.edu*

## 1. Overview

Services in the digital ecosystem operate on a steady stream of users' personal information. While each piece of information has a legitimate use in providing services the user wants, its use in other contexts may be improper: user privacy is maximized when each service has access only to the information it needs. However, most services are provided by a small set of very large entities, either corporations or governments. These entities can alter their service provision to extract value based on subjects' information: their utility is maximized when each service has access to all information. Maximal information usage enables surveillance capitalism in the corporate case, and the surveillance state in the government case. Surveillance research has warned about single information collections with no legitimate use, and specific illegitimate uses of a collection; but it has not provided a way to systematically detect and prevent misuse of subject information that is otherwise surrendered consensually. We propose to solve this by introducing *surveillance graphs*, which represent the whole flow of information about users within a surveilling entity. Analysis of surveillance graphs shows each case where processes have access to information they do not need: if such a case exists, the graph is called *entangled*. An entangled graph may have a *disentangling cut*, which leaves all services intact but not entangled, maximizing user utility and privacy. Disentangling cuts are proposed for applications in corporate responsibility and democratic reform.

## 2. Surveillance Graphs

This research proposes a graphical model for representing an entire system of surveillance: a *surveillance graph* consists of vertices which are either observations, identities, or responses. Consider a small surveillance system in which a red-light camera observes cars running a red light, and the driver is sent a fine as punishment. We call the camera recording of the car running a red light an *observation*. Directly available in the recording is the car's license plate: this is an *identity*. However, the fine cannot be sent directly using the license plate. A government database connects the license plate number to a name and address: these are additional identities. Finally, the fine is sent directly to the name and address: the sending of the fine is a *response*, because it is contingent upon the observation (whether a given car ran a red light or not). By definition, a directed edge from a tail vertex to a head vertex exists when information about the tail vertex can be routinely associated with the head vertex.

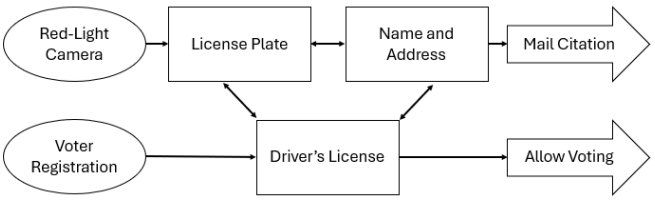



Figure 1. A simple hypothetical surveillance graph. Red-light camera recordings can be associated with the subject's voting, so this surveillance graph is entangled. Where an edge is shown with arrow heads on both ends, an edge exists going each way.

## 3. Entanglement and Disentangling Cuts

Sometimes a surveilling entity is able to access information during a response which is not appropriate for that response. In Figure 1, we augment the surveillance system described in Section 2 so that the identities of license plate and name and address are also connected to a driver's license. The driver's license, in turn, is used for voters' registration and for voting. In this case, the surveilling entity has the ability to consider the subject's voting registration information when deciding whether to issue a ticket. However, while voter registration information is required to correctly allow voting, red-light camera recordings are not required; use of red-light camera recordings are inappropriate. Because this privacy violation is possible, the surveillance graph is *entangled*.

An entangled surveillance graph can be *disentangled* by removing the edges in a *disentangling cut*. We cannot cut all edges: in this case, if the system cannot connect voter's registration to voting, it cannot correctly provide voting as a service. We say that the observations required to properly provide a response are the *service set* of that response.

If every observation in the service set of a response can be associated with that response, and this is true for all responses, we say the graph is *intact*. A disentangling cut is a set of edges which, when removed from the graph, leave the graph still intact but no longer entangled. In Figure 1, a simple disentangling cut exists: {License Plate → Driver's License, Driver's License → License Plate, Name and Address → Driver's License, Driver's License → Name and Address}. If these edges are removed, the surveillance graph will remain intact but will be disentangled.

## 4. Disentangling Cut Algorithms

The general disentangling cut problem on directed graphs is NP-complete by reduction from the subgraph homeomorphism problem [1]. For experimentation, we generate a set of synthetic surveillance graphs: by construction, all graphs have known disentangling cuts, and each response has only one observation in its service set. We test three algorithms for computation time and success rate. The first algorithm is exhaustive: all subsets of edges are tried, starting with the smallest subsets, until a disentangling cut is found or a time limit (10 seconds) is reached; this times out on most graphs. The second algorithm is naïve: a breadth-first search from each observation to its paired response finds a shortest path, and if all responses' paths are disjoint, the edges between them are taken as a disentangling cut: this is very fast but fails on most graphs. The third algorithm uses "squeaky wheel" escalations in priority: services claim paths from observations to responses, but if a path is unable to be found, the priority for that service will be elevated and it will be able to take over vertices claimed by other services; this achieves high (but not 100%) success in moderate time.

## 5. Public Surveillance Documentation

Many companies with large surveillance systems disclose the information they collect and the purposes for which that information are used. This reveals a subgraph of the surveillance graph: many vertices (observations, identities, or responses) are disclosed, and the existence of some paths (from observation to response, or from identity to response) are disclosed, but individual edges are not. Because the disclosed subgraph is not intact, the subgraph may be supposed to be incomplete.

For demonstration, a fictional surveilling entity ("MOOGUL") is hypothesized, and a surveillance graph is created for this entity starting with the disclosures about two services of a non-fictional company (Google Search and Google Pay). The disclosures are mapped on to a subgraph of the surveillance graph, and then the graph is augmented so that it is intact according to the service definitions of the non-fictional disclosures. The MOOGUL surveillance graph is large, but part of the graph is shown in thumbnail form in Figure 2. This surveillance graph is entangled, but does not contain disentangling cuts.

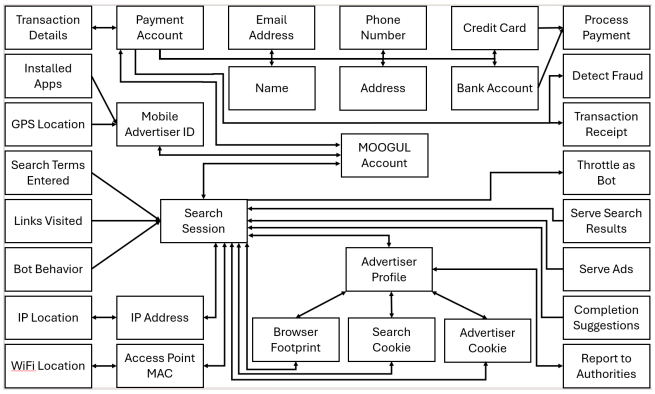



Figure 2. A small part of the surveillance graph for MOOGUL, a fictional company which collects large amounts of subject data. This graph includes MOOGUL's internet search and internet payment services.

## 6. Discussion

Implementation faces two key challenges. The first is that surveillance graphs are generally not public, and both kinds of surveilling entities have an interest in keeping them unknown. Private companies' surveillance graphs may disclose their system design, risking their competitive advantage; governments' surveillance graphs may reveal the possibility of privacy violations, risking popular opinion.

Second, many surveillance systems will be designed to be minimal and efficient, and therefore have no disentangling cut. These graphs must be modified to allow for disentangling cuts to exist. One way to modify graphs to allow disentangling cuts is to duplicate identity vertices which are required for multiple services, allowing different services to use different copies of the vertex, and cutting different edges on different copies of the vertex.

Disentangling cuts on surveillance graphs have two key applications. First, private companies' surveillance graphs may be disentangled to preserve customer privacy and marketplace fairness, and can be motivated by government privacy regulation. Second, democratic governments' surveillance graphs may be disentangled by elected government officers with a political philosophy that limits government intrusion into the lives of subjects, if they are elected.

Even before before best practices dictate disclosure of surveillance graphs, or regulation imposes requirements on them, companies and governments can internally analyze their surveillance systems and use the graphs as proof that routine misuse is not possible (because discriminatory information is not available at point of decision for sensitive responses).

## References

[1] S. Fortune, J. E. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem," *Theor. Comput. Sci.*, vol. 10, pp. 111–121, 1978. [Online]. Available: https://api.semanticscholar.org/CorpusID:30803478

# Disentangling Cuts on Surveillance Graphs


**William Schulze**
Michigan State University
Computer Science and Engineering
schulzew@msu.edu

MICHIGAN STATE UNIVERSITY

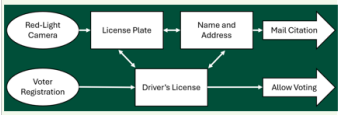
**Sandeep Kulkarni**
Michigan State University
Computer Science and Engineering
sandeep@msu.edu


## 1. Surveillance Systems

- Passively collect **observations**
- Associate them with various forms of **identity**
- Use them to decide how to apply **responses**
- Are opportunistic, passive, and systematic

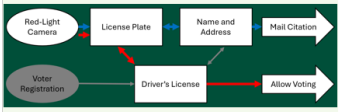

## 2. Past Research

- Contextual Integrity assumes joint negotiation of context: not possible in surveillance
- Other study single problematic observations or single problematic responses
- But these usually have some justification
- Can we reform information misuse while preserving popular or justifiable uses?
- Graphs can help us analyze the whole system



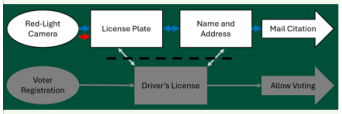

## 3. Surveillance Graphs

- **Intact**: paths exist from needed observations to responses
- **Entangled**: paths exist from unnecessary observations to responses




## 4. Disentangling Cuts

- A set of edges to **remove** from the graph
- Graph must remain intact but no longer be entangled
- **Not all graphs** have disentangling cuts




## 5. Real-World Surveillance Graphs

- Both **corporate** and **governmental** entities have an incentive for their graphs to remain unknown
- Public pressures government, and government pressures corporations, to **disclose**
- Disclosures are **partial**, but can be **augmented** into a plausible hypothetical graph
- Manual augmentations are **more plausible**
- Algorithmic augmentation is unaffected by **researcher bias**

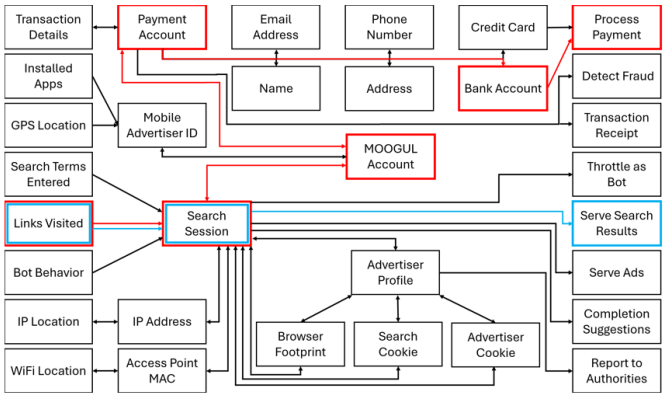
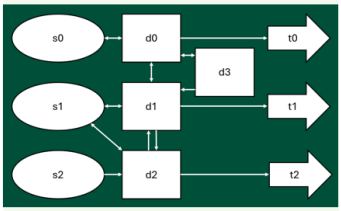



Figure 1: a surveillance graph for "MOOGUL" (Massively Online Observational and General-Utility LLC), a fictionalized company. The MOOGUL surveillance graph starts based on Google's public disclosures and is augmented by adding **plausible, but not documented**, vertices and edges until intact. The "Links Visited" observation is required for "Serve Search Results" (**blue path**); this is required for the graph to be intact. However, it is also available to "Process Payment" (**red path**); this makes the graph entangled. The identity "Search Session" is common to both paths.

## 6. Synthetic Graphs

- Useful for **algorithm testing**
- Graphs with a **known** disentangling cut
- Graphs modeling a **redundant** design
- Graphs modeling a **minimalist** design
  - Unlikely to have a disentangling cut




## 7. Disentangling Cut Algorithms

- Problem is **NP-complete**
- **Greedy** algorithm is fast, but often fails
- **Exhaustive** algorithm runs in EXPTIME
- **Squeaky-wheel** is polynomial and has good, but not guaranteed, success

## 8. Application

- Require surveilling entity to disclose surveillance graph
- Test for entanglement
- Calculate disentangling cuts and remove links in the surveillance system

## 9. Key Open Questions

- How should **disclosures** be translated into vertices and edges? What edges and vertices should be added in graph **augmentation**?
- What **properties** do real-world surveillance graphs have? How can these aid **algorithms**?

## 10. Contributions

- **Systematic** analysis of surveillance systems
- Discover privacy violations at **several steps' remove**
- Prove that systematic privacy violations are **not possible** in a system
- **Restore** privacy while **preserving** popular services