Privacy Preserving Visualization for Social Network Data with Ontology Information

Jia-Kai Chou* Chris Bryan† Kwan-Liu Ma‡

University of California, Davis

ABSTRACT

Analyzing social network data helps sociologists understand the behaviors of individuals and groups as well as the relationships between them. With additional ontology information, the semantics behind the network structure can be further explored. Unfortunately, creating network visualizations with these datasets for presentation can inadvertently expose the private and sensitive information of individuals that reside in the data. To deal with this problem, we generalize conventional data anonymization models (originally designed for relational data) and formally apply them in the context of privacy preserving ontological network visualization. We use these models to identify the privacy leaks that exist in a visualization, provide graph modification actions that remove and/or perceptually minimize the effect of the identified leaks, and discuss strategies for what types of privacy actions to choose depending on the context of the leaks. We implement an ontological visualization interface with associated privacy preserving operations, and demonstrate with two case studies using real-world datasets to show that our approach can identify and solve potential privacy issues while balancing overall graph readability and utility.

1 INTRODUCTION

With recent advances in information technology and social networking platforms, person-to-person interaction data is now widely collected and readily available. Domain researchers such as sociologists use this data to better understand the behavior and interaction patterns within and between populations. One way to present and analyze this kind of data is with an ontology graph, which specifically denotes the different types or categories (*i.e.,* ontologies) of nodes within the graph. In this way, the semantics associated with a social network can then be better understood [23].

Network visualization can effectively show the complex relational concepts contained in this type of data. Unfortunately, the existence of categorical groups, data extremas, and set intersections can allow for nefarious and potentially hostile attackers to uncover, either directly or indirectly, a specific individual's personal information; this knowledge can then be leveraged for ulterior intents. Studies have shown that sensitive information in a social network can be de-anonymized purely by analyzing its topology with some auxiliary information, *e.g.* node degree, thus violating the privacy and anonymity of its members [2,20]. When a social scientist wishes to prepare a graph visualization for sharing of findings (either with colleagues or to the general public), s/he may inadvertently expose sensitive personal details that reside in the data.

With these concerns in mind, privacy preservation approaches have been developed which prohibit data mining techniques from being able to identify individuals in social networks. These mostly operate at a data- or algorithmic-level (and are mostly designed for non-ontological data schemas). In contrast, little effort has been put forth in the field of visualization, which places an emphasis on perceptual-based discovery, analysis, and cognition, and allows for a graph builder to interactively decide which data elements can be considered salient (and which privacy leaks can safely be ignored).

In this paper, we discuss privacy issues that arise when visualizing ontological networks using graphs and adjacency matrices. To detect privacy leaks in networks, we leverage relational data models. These have an advantage over purely topological models in that they emphasize the categorical types of connections between nodes in the data (*i.e.*, between the different ontologies). Using these models (specifically, *k-anonymity* [25] and *l-diversity* [19]), we show how to detect leaks and suggest graph augmentations that "fix" detected leaks; these mostly operate at a data-level in that they modify the actual topology of the graph. We also discuss how leaks can be perceptually mimimized (*i.e.*, modifying the appearance of the graph to cognitively "mask" the leak's appearance) in cases when a data-level augmentation is not the most favorable option.

To assist with choosing the appropriate corrective actions, we suggest strategies based on the type of leaks that are detected in conjunction with the set of nodes and/or edges they affect. Based on these, we develop an interactive prototype system for ontological network visualization building and privacy preservation. Its workflow is straightforward: (1) Find leaks in the graph. (2) Perform augmentations to obfuscate or alleviate the issues. (3) Refine the look of the graph to remain suitable for presentation.

By letting a user iteratively build and anonymize an ontological graph visualization, s/he can intuitively balance between privacy and utility (that is, semantic readability) both before and after privacy operations are performed on it. This lets the user ensure important leaks are either removed or minimized, while maintaining an end result that is still suitable for presentation. To evaluate, we demonstrate two case studies showing how the discussed techniques work in a privacy preserving, graph building process. Feedback from interviews with sociologists who work with sensitive, social network datasets is also discussed.

2 RELATED WORK

Prior related work can be discretized as such: (1) privacy preservation for network data, (2) ontology-based network visualization, and (3) generalized privacy preserving visualization techniques.

2.1 Preserving Privacy in Social Network Data

The problem of privacy preservation has been extensively studied in the data mining field. For relational data in particular, prior research has established notable generalized anonymization models such as *k-anonymity* [25] and *l-diversity* [19]. These serve as a basis for many network data mining approaches, though some include a focus on leveraging other topology metrics such as node degree, neighborhood information, shortest path, edge weights, and entity grouping [4,6,18,35,36].

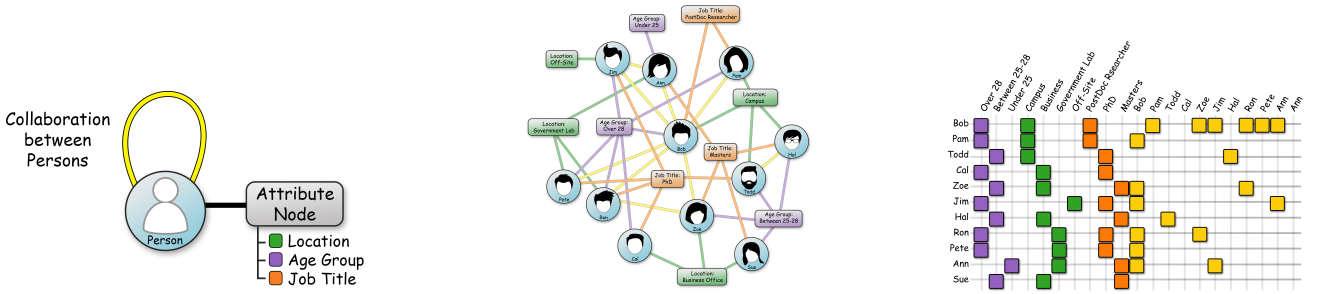
A survey by Zhou *et al.* [37] notes two commonly-used anonymization techniques, *i.e.*, procedures that can transform a privacy leaking graph into one that is privacy preserving: (1) merging

*e-mail: jkchou@ucdavis.edu

†e-mail:cjbryan@ucdavis.edu

‡e-mail:ma@cs.ucdavis.edu

2017 IEEE Pacific Visualization Symposium (PacificVis)
18-21 April, Seoul, Korea
978-1-5090-5738-2/17/$31.00 ©2017 IEEE



(a) Example Ontology Schema (b) Showing with Node-link Diagram (c) Showing with Adjacency Matrix

Figure 1: Illustrating a synthetic, ontological social network of eleven researchers affiliated with a university lab. (a) Based on an example schema, entities are of type **[Person]**, and connections between them show **[Collaboration]**. There are three types of demographic attributes: **[Location]**, **[Age]**, and **[Title]**. (b) The data can be laid out in a force-directed graph. Each **[Person]** node has an edge to its respective set of attribute nodes. (c) The network can alternatively be shown in an adjacency matrix.

nodes/edges and (2) adding/deleting nodes/edges. We leverage both actions in this paper. Node merging is also used by [31] and [34] as a way to achieve *l-diversity*; the latter additionally considers graphs with multiple types of edges, which can be deleted based on their ontology. Similar to our merging approach, Hay *et al*. [13] present a generalization technique that groups individual nodes into "super-nodes" and edges into "super-edges" to address *k-anonymity*.

Randomization is another approach to preserving privacy, in that random node/edge deletions/insertions can inject a measure of uncertainty (and potential dis-truthfulness or error) into the data [14,17]. Similarly, edge swapping randomly replaces endpoint connections [32], and noise nodes add additional elements to the display [30]. Though these techniques have been used in the sociology community when presenting sensitive social networks, the false information introduced might be costly (see Section 7.1). Therefore, we do not consider randomness as a viable approach in this paper.

A constraint to many of these works is that they achieve graph anonymity purely via theoretical approaches and solely modify the graph at a data-level, sans user discretion. In a visualization context, more nuance may be required; a graph designer can interactively consider how a privacy leak should be dealt with or whether a privacy leak may be allowed based on its context. Instead of removing the leak at a zero-tolerance level (*i.e.*, by an action like node merging or randomization), it sometimes may be more desirable to perceptually minimize the visibility of a leaking node. While still existing at a data-level, it is hidden from easy recognition. Additionally, when building a graph for presentation, the semantics of the network must be taken into account. The chart should communicate appropriate insight to the viewers. Few prior papers discuss these aspects of visual reasoning when a dataset is analyzed.

2.2 Ontology Graph Visualization

Visualizing heterogeneous social networks with ontology information is an effective technique for social network analysis. For example, Shen *et al*. [22,23] use ontology information to semantically prune or reduce the size of inherently large and complex networks. This makes the task of inferring important relationships and revealing hidden knowledge within a graph more manageable. At a cognitive level, Oellinger and Wennerberg [21] point out that a major advantage of included ontology information is the deployment of inference mechanisms and the possibility to extend and refine the network with further (sub)concepts.

Many recent visualization papers have looked at improving readability in ontological graphs. These include specialized layouts based on domain ontology models to emphasize social structure [27, 28], grid-based displays to emphasize edge saliencey [26], force-directed displays that leverage information about node degree distributions [16], and node modification strategies such as merging to aggregate hierarchical nodes [24], representing ontology-based cliques (or clusters) with adjacency matrices [1], and adopting mixed-initiative approaches that allow both automatic force layout with additional manual adjustment [33]. This last paper is similar to our approach, in that our system allows a user the ability to apply different graph layout settings and then tweak, pin, and drag individual nodes to achieve a desired look and positioning.

2.3 Privacy Preserving Visualization

Recently, privacy preservation has received increased emphasis in the visualization community. Examples include applying *k-anonymity* and *l-diversity* to parallel coordinates [7], investigating privacy issues in event sequence datasets [5], and discussing opportunities and challenges for privacy preserving visualization in the realm of electronic health record data [8]. In general however, supporting visual analytics tasks on various types of privacy-sensitive data is lacking, and should be further investigated.

3 Visualization of Ontological Social Networks

This section formally defines ontology networks and how they can be plotted with node-link diagrams and adjacency matrices. We discuss layout considerations for both chart types, and how different approaches to plotting ontological data (node-link diagrams vs. adjacency matrices) can affect the perceptual readability and semantical understanding with regards to the underlying data.

For clarity, the terms *node* and *vertex* are used interchangeably for node-link diagrams. *Cell* is used for adjacency matrices. The term *entity* has a special connotation; it is the main ontology-type for the graph. Since this paper focuses on social networks, an entity then always refers to a person. To ease the discussion and reduce potential confusion, we consider the edges or connections of the datasets mentioned in this paper to be undirected.

3.1 Defining Ontological Social Network Data

We define ontology graphs following the notations in [22, 23]. Let a network graph be denoted as $G = (V, E, vt, et)$ and its associated ontology information defined as $OG = (T_V, T_E)$. V and E are the set of vertices and edges in the graph, respectively. $T_V = \{t_1, t_2, ..., t_m\}$ is a set of categorical vertex types and $T_E = \{(t_i, t_j) : t_i, t_j \in T_V\}$ is a set of edge types. vt denotes a mapping from V to T_V that associates a vertex to its type. That is, for a vertex v in the graph, $vt(v)$ refers to the type of the vertex v. Similarly, et denotes a mapping from E to T_E that associates an edge with its type.

3.1.1 Organizing Ontology with a Schema

In a sense, an ontological social network can be considered as a basic, person-to-person network that has been augmented with the addition of attribute-specific types of nodes that are linked to the initial set of person-based nodes. The ontology information of a social network (*OG*) explicitly specifies the nature of entities and relations that exist in this modified or enhanced network.

The allowable connections in a social network can be defined by a schema. Figure 1(a) shows a schema for a synthetic network of eleven researchers plotted in Figure 1(b-c). This example schema defines four types of nodes: [Persons] (which are the entity nodes), [Locations], [Ages], and [Titles] (the latter three are attribute nodes). Edges between nodes of differing ontologies show a type-specific relationship between the nodes. For example, a [Person]-to-[Location] edge signifies where the person works. Depending on the graph's contextual rules, a node may have a one-to-one, one-to-many or even many-to-many relationship with other nodes. For example, a person can only be at one [Location] at a time, but a location can have many [Persons] working there.

An edge that traverses two vertices with the same ontology can be used to show mutual connection. In the schema, a yellow edge connecting two people indicates they collaborate together. If the schema includes hierarchical ontologies, this can be shown by edges between attribute nodes of the same type [24]. (Figure 1(a)'s schema does *not* include this.) For example, if larger, "city-level" [Location] nodes are included in the dataset, an edge from a work-specific to a city-level [Location] would indicate the next-level up in the locational hierarchy.

3.2 Visualizing Ontological Social Network Data

The two most common techniques for visualizing social networks are with graphs (*e.g.*, node-link diagrams) and adjacency matrices.

It is straightforward to augment node-link diagrams to include ontological information, as shown in Figure 1(b). Based on schema swatches, color differentiates ontologies. The color of a node maps to its type, and the color of an edge maps to the relationship ontology between its endpoint nodes. (Alternatively, though node/edge channels such as shape, alpha, and transparency can instead be used to show ontology attributes, this limits the scalability of attributes that can be shown.) Here, people are shown as blue nodes; yellow edges show their collaborative relationships. Green nodes show the location where a person works; green edges connect each person node to their location. Similarly, purple and orange nodes/edges identify age groups and job titles.

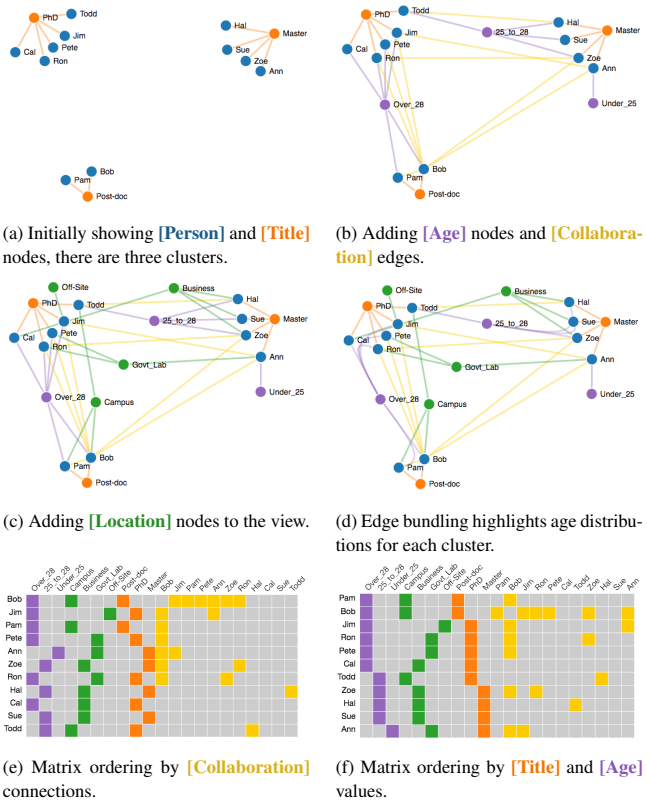
A common alternative to node-link diagrams are adjacency matrices (Figure 1(c)). While a traditional social network (sans ontology) with n people can be represented by an $n \times n$ matrix, to include ontological information extra columns must be added. The dimensions of the resulting matrix becomes $n \times (m+n)$ where m denotes the number of ontology types in the network. Similar to node-link diagrams, cell color can indicate connection value and ontology between two nodes.

3.3 Improving Ontology Visualization Readability

Successful visualization should allow viewers to effectively perceive, analyze, and interpret an underlying dataset and its salient features. There are several considerations for this in the context of ontological social networks; we consider three primary ones: (1) choice of graph layout (*i.e.*, positioning nodes to emphasize certain semantics), (2) use of edge bundling (alleviating clutter by bending edges together), and (3) matrix row/column reordering (to highlight patterns or groups within the data).

3.3.1 Iterative Ontological Graph Layout

For ontological node-link diagrams, force-directed layouts are frequently used [22, 23, 27, 28] due to their simplicity, flexibility and



(a) Initially showing **[Person]** and **[Title]** nodes, there are three clusters.

(b) Adding **[Age]** nodes and **[Collaboration]** edges.

(c) Adding **[Location]** nodes to the view.

(d) Edge bundling highlights age distributions for each cluster.

(e) Matrix ordering by **[Collaboration]** connections.

(f) Matrix ordering by **[Title]** and **[Age]** values.

Figure 2: Techniques for allowing ontological semantics to be more easily perceivable. (a)-(c) Iterative graph building by adding nodes and edges. (d) Edge bundling. (e)-(f) Matrix reordering.

generally pleasing results. However, naive implementations usually do not consider node ontologies. This can make it more difficult to interpret the semantical relationships between multiple entity and attribute nodes. Figure 1(b) shows this, as nodes are placed to minimize all edge lengths without considering their ontology types.

One solution to this issue (adopted both by [33] and our prototype system) is to use a mixed-initiative approach. An automatic layout determines placement for an initial set of nodes and edges. The user then refines the graph through multiple, successive stages of adding nodes/edges, filtering out undesired ones, and tweaking their placements by dragging and pinning nodes. A key benefit to this approach is flexibility– the user can optimize the graph presentation to highlight its semantics, especially when multiple attributes of the network are presented at the same time.

This approach also gives full control over which specific ontological attributes from a datset should be included for display. Only necessary information is presented, which reduces graph clutter. While iterative refinement of a large graph in this way can lead to scalability issues, effective use of data clustering, layouts, and semantic/structural filtering can help address this, see [16, 23]. A thorough discussion of this topic, however, is beyond the scope of this paper.

Figure 2(a)-(c) demonstrates the iterative building process used in our prototype system. Upon loading the initial dataset (the same as in Figure 1), only blue [Person] and orange [Title] nodes are displayed. (a) These nodes are arranged to show the three distinct groups in the data and pinned to the display. (b) Next, purple [Age] nodes are added, along with yellow [Collaboration] edges. (c) Finally, green [Location] nodes are added. At each stage nodes are arrayed according to the user's preference for highlighting certain

data semantics. New nodes are positioned using force-direction, and can then be pinned or adjusted.

3.3.2 Edge Bundling

To improve legibility, edge bundling is an aesthetically pleasing technique that simplifies graph complexity and better shows the structure or patterns that edges can have [15]. Edge bundling algorithms reduce visual clutter by pulling adjacent edges together, transforming them from straight lines into curved splines that fan out near the termini of their respective nodes. While edge bundling was originally introduced as a way to group node sets together in hierarchical data graphs (*i.e.*, bundling edges together at each hierarchy level), the same concept can be extended to categorical edge types found in ontology data. For example, Figure 2(d) shows a continuing example from Figure 2(a)-(c), where the edges for purple [Age] nodes have been bundled together for different clusters.

3.3.3 Matrix Reordering

Row and column order of an adjacency matrix can intuitively show high-level data patterns, such as clusters or highly-connected nodes [3]. This ordering benefit becomes more prominent for larger and denser networks. Figure 2(e) shows rows and columns ordered by collaboration links between people. Compared to Figure 2(f), it is more straightforward to see that Bob (the first row) is highly collaborative with others while two people (Sue and Zoe) do not collaborate at all.

Ordering can also be done based on ontology attributes. Figure 2(f) reorders the matrix by [Title] groups, which highlights the three clusters of people similar to the graph layout in Figure 2(a).

4 Privacy Considerations in Ontological Social Network Visualizations

Here, we describe how two privacy models that are commonly used in relational data mining can be leveraged to identify privacy leaks in ontological social networks. To "fix" a detected leak at the data-level, graph modification operations that change the topological structure of the graph or matrix are applied. In some cases, these actions are not always the most favorable solution. An alternative is changing the appearance of the visualization in a way that perceptually "hides" a leak. While still present in the graph, it becomes harder for a viewer to perceive at the cognitive level. Based on the types of leaks that are found and the sets of nodes/edges that they affect, we recommend strategies for automatically performing graph modification actions.

4.1 Privacy Models for Ontological Social Networks

At a data-centric level, prior studies have introduced various anonymization techniques for social network data. Unfortunately, defining what should be the standard or best practice for privacy leak detection remains an open question. Multiple researchers [20, 29, 31, 37] have emphasized the impracticality of expecting one single anonymization approach being able to address *all* forms of privacy issues, or even being able to catch and preserve all leaks within a network: *"Generally, graph data is sufficiently complex that it is impractical to prevent all forms of disclosure with a single anonymization approach."* [31] Because of this, one must make assumptions about what types of leaks are important and thus worth fixing, based on the context of the data.

The issue is further complicated when ontology is considered, as many social network approaches do not consider nodes and edges of differing types. While certain ontology types may be considered *sensitive*, that is, able to expose the privacy of a particular entity node (person) or wished to be kept hidden, others ontologies may not be. Even within sensitive ontology types, specific nodes may not be regarded as sensitive.

Based on these assumptions, privacy detection approaches designed for relational data (as opposed to only social network data) can be leveraged and applied in the new context of ontological social networks. There are two relational-based privacy models that are particularly apt for this process: *k-anonymity* [25] and *l-diversity* [19], which we discuss here, though other social network-specific approaches can also be used (see Section 7.2).

4.1.1 Defining *k-anonymity* and *l-diversity*

Formally, *k-anonymity* is defined such that each equivalence class contains at least k records, therefore any single record in the same equivalent class cannot be distinguished from the other $k-1$ records. Applying this in an ontological social network context, an equivalence class is formed by a group of entity nodes who are linked to a single or a set of common attribute nodes.

For example, in Figure 1(b), there is only one [Person] (Jim) linked to the Off-Site [Location]. Jim's 2-anonymity is violated (since less than two people are connected to Off-Site). If an attacker then knows that Jim works at Off-Site, s/he will also know Jim's [Age] (Over 28) and [Title] (PhD), thus Jim's privacy has been entirely compromised.

l-diversity extends the concept of *k-anonymity* by requiring the records contained in each equivalence class to obtain at least l different sensitive attributes. In Figure 1(b), both Bob and Pam are linked to the PostDoc Researcher [Title] node. This satisfies *2-anonymity* criteria, but violates *2-diversity*, as both Bob and Pam also are linked to Campus [Location] and Over 28 [Age]. Since they both share the same job title, and because their other attributes map similarly, it can be identified that a PostDoc researcher in the graph is definitely located at the Campus and is over 28 years old, violating both Pam and Bob's privacy.

4.2 Privacy Preservation Actions

To resolve *k-anonymity* and *l-diversity* privacy leaks, we consider three types of graph modifications: (1) node/edge deletion, (2) node merging, and (3) edge bundling. These actions change the topological structure and the underlying data representation of the node-link diagram (and the adjacency matrix) in a way that removes the leak at a data-level. Unfortunately, as this reduces the overall utility of the visualization, it means that each action has a cost associated with it. In discussing this, we use a five person sample dataset shown in Figure 3 (using the same ontology schema as Figure 1(a)).

4.2.1 Node and Edge Deletion

In a node-link digram, deleting a node removes it and the edges linking to it. Deleting an edge similarly removes a connection between two nodes. The equivalent deletion action in an adjacency matrix removes the appropriate row/columns. If the node is an entity (that is, a person), then both a row and a column is removed. If an attribute type of ontology node is removed, only a column needs to be removed. Deleting an edge in an adjacency matrix removes only that particular cell's value.

Node deletion is a particularly effective operation to use when a privacy leak is "isolated" or independent of connections to many other nodes. For example, in Figure 3(a), there is only one [Person] node (Dave) who is a Rookie. If the graph is presented with personal names hidden, but an attacker already knows Dave is a Rookie, then s/he can derive Dave's other (potentially sensitive) information (*i.e.*, he is an Engineer who's workplace is Home).

Deleting the Rookie node fixes this isolated privacy leak. However, this type of operation should be applied with caution, as it removes information entirely from the graph as opposed to merely introducing uncertainty through obfuscation.

4.2.2 Node Merging

Node merging combines two or more nodes into a single "super-node." Edges connected to these nodes are now aggregated to the new node, while edges going between the merged nodes are hidden (inside the super-node). In conventional node-link diagrams, this technique is often used to organize nodes according to a data hierarchy, especially in the context of graph simplification (reducing node/edge density) and interactive exploration [10].

Considering privacy, a merged node obfuscates privacy leaking information by adding uncertainty to the set of merged nodes. A viewer is unable to tell which edges connecting to the super-node go to which specific node inside. However, care must be taken if nodes of different ontologies are allowed to be merged, as edge types and/or weights can vary. In our current prototype system (see Section 5), we restrict node merging to sets of the same ontology and with the same edge weight.

Node merging is especially suitable for addressing *k-anonymity* leaks, which are caused by insufficient node degree (the node degree is smaller than the value of k). The visual effect of node-merging in an ontological network is the same as in a conventional node-link diagram (see Figure 3(b), where the privacy leaking Rookie node is merged with the 1+yrs node to fix a $k = 1$-anonymity leak). For an adjacency matrix, when nodes are merged their columns are also merged. In our system, we expand the size of the columns corresponding to the number of merged nodes to help indicate the merged node's aggregated size (Figure 3(c)).

4.2.3 Edge Bundling

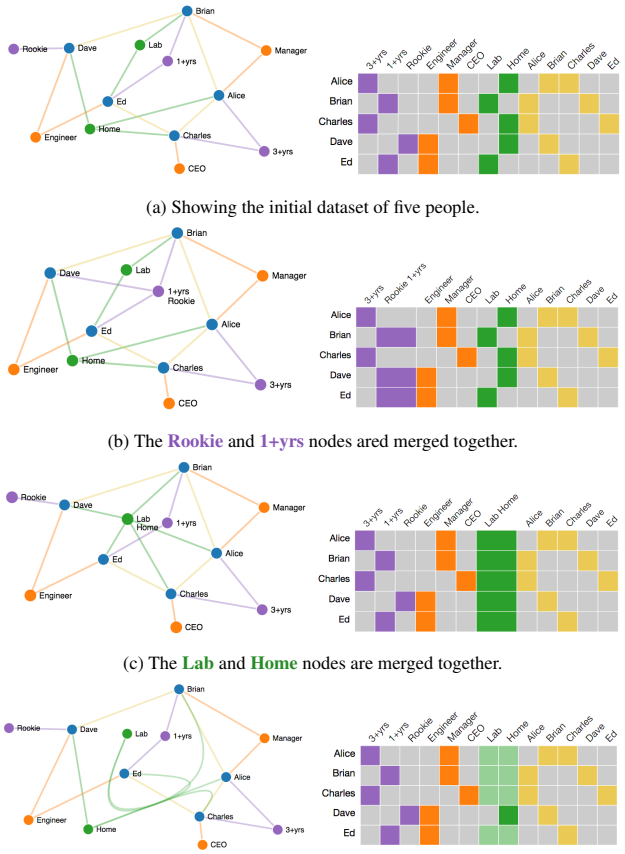
Edge bundling as a readability-improving and clutter-reducing technique is discussed in Section 3.3.2, but it can also be purposed to preserve privacy. Tight visual bundling obfuscates the specific source and destination end points for a set of edges between two groups of nodes. This prevents a viewer from telling where the nodes from one group specifically go to in the other group. In Figure 3(d), a set of [Person] nodes have their edges bundled going to green [Location] nodes. It is apparent that each person is attached to a location, but a viewer cannot tell which explicit location they map to (either Lab or Home). The adjacency matrix shows edge bundling by duplicating cell values, with half-opacity denoting the uncertainty of the bundled edges.

Edge bundling is particularly suitable for addressing *l-diversity* leaks which can be caused by having certain sets of edges linking to the same source and destination nodes. In Figure 3(a), a *2-diversity* leak happens because both Charles and Alice work at the Home [Location] and have 3+yrs of work [Experience]. If node merging is applied to combine the Home and Lab nodes (Figure 3(c)), then the Dave node's working location is obscured. Instead, by edge bundling the set of edges in Figure 3(d), enough information is obfuscated to fix the specific privacy leak while keeping other edges intact so that their patterns can be preserved. The information that Dave's working location is Home [Location] is still preserved while the information of other nodes' working locations is obfuscated.

4.3 Perceptually Masking Privacy Issues

Although privacy leaks can be computationally detected at a data-level, zero-tolerance graph modification actions may not always be the best desirable solution. For example, removing and aggregating too much information from the graph or adjacency matrix can reduce its overall usefulness. An alternative choice then is to perceptually minimize (that is, to hide) existing leaks from easy identification by a viewer.

In node-link diagrams, edge crossings, naive node placement, long edge lengths, and node/label overlapping make graphs less readable and thus more difficult to interpret. This goes directly against improving the readability of a visualization, as discussed in Section 3.3, but paradoxically can serve to mask privacy leaks



(a) Showing the initial dataset of five people.

(b) The **Rookie** and **1+yrs** nodes ared merged together.

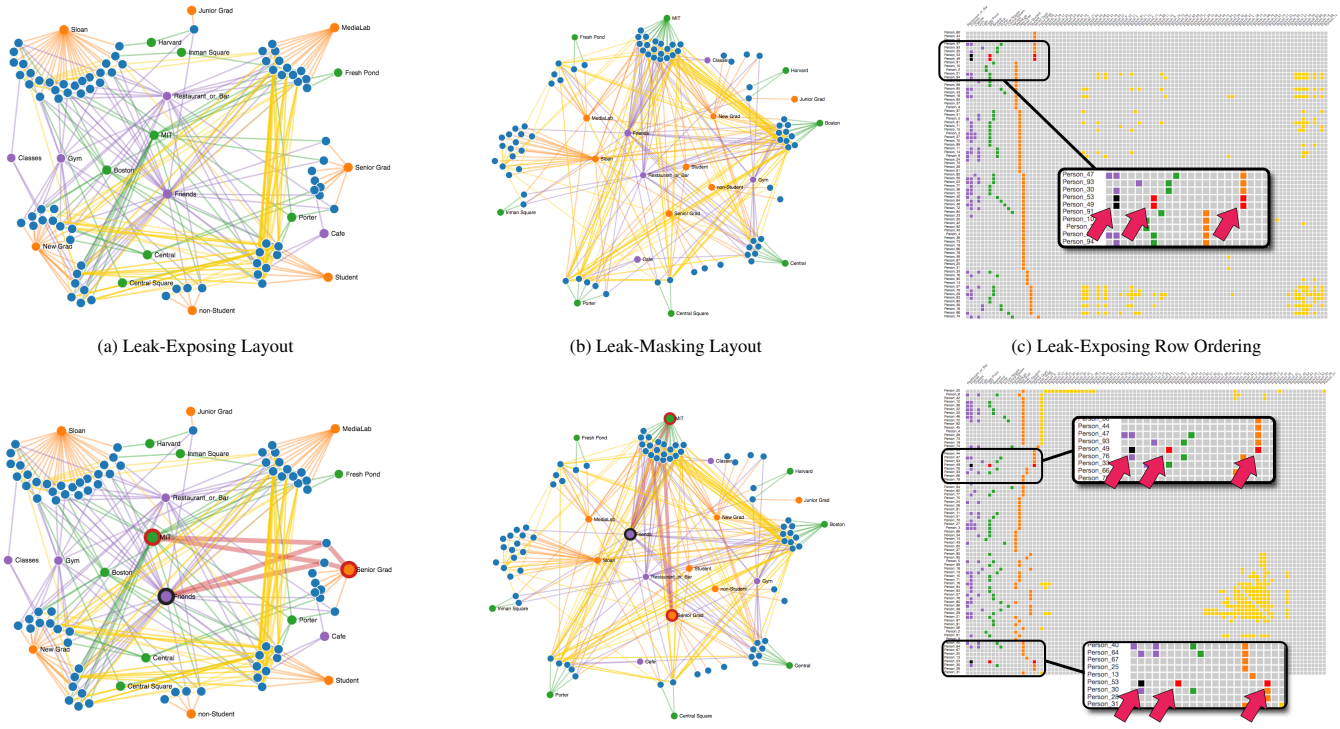
(c) The **Lab** and **Home** nodes are merged together.

(d) Edge bundling **[Person]** nodes connected to the **Lab** and **Home** nodes.

Figure 3: Privacy preserving operations: (a) The original dataset. (b) Applying a node merge to address k-anonymity. (c) Applying a node merge to address l-diversity. (d) Edge bundling to address l-diversity.

by making it harder to isolate nodes or entity combinations that are sensitive and exposed. For example, Figure 4(a) and (b) shows two different layouts for a dataset. We highlight a privacy leak in these two respective layouts in Figure 4(d) and (e). In Figure 4(e) the two highlighted person nodes are positioned in a way that their connections are "shielded" by clustering with other nodes. The edges that lead to the privacy leak heavily overlap with other, non-privacy leaking edges. In contrast, in Figure 4(d), the connections (and thus the privacy leak) for these two persons are easily identifiable: two Senior Grads who live in MIT both go to the hangout place Friends, more easily– a leak that violates 2-diversity.

When considering the perception of privacy issues in ontological adjacency matrices, the biggest perceptual cue for a viewer is the arrangement, sparsity, and the dispersity of the cells that expose sensitive information. This is a result of the row/column ordering of the adjacency matrix, as Section 3.3.3 notes. Reordering the rows and columns of an ontological adjacency matrix can reduce the dispersity of certain cells and more clearly indicate certain relationships or semantics between the sensitive cells, but at the same time make certain patterns more scattered and harder to reason. In Figure 4(c) the privacy leaking cells are placed on successive rows in the matrix which makes the privacy issue easier to spot. The same privacy issue is highlighted in Figure 4(f), but is perceptually more difficult to recognize as the offending cells are now in rows that are far apart from each other. As a result, row and column reordering should be carefully done in a way that balances exposing patterns



(a) Leak-Exposing Layout

(b) Leak-Masking Layout

(c) Leak-Exposing Row Ordering

(d) Leak-Exposing Layout, Leak Highlighted

(e) Leak-Masking Layout, Leak Highlighted

(f) Leak-Masking Row Ordering

Figure 4: Graph layouts and matrix ordering affect privacy perception. All six plots show the same dataset, which contains the following privacy leak: two Senior Grads who live at MIT both hang out at Friends, a 2-diversity leak. (a) and (b) show two layouts of the graph with the leak unmarked (graphs (d) and (e) show the same layout with the leak highlighted). In (a), the leak is more visually perceptible, while in (b) it is hidden by line clutter and node positioning. In (c), the matrix row ordering places the leak-causing nodes by each other, allowing easier recognition as opposed to (f), where they are placed far apart in the set of matrix rows.

within the data while minimizing exposure of non-essential privacy leaks.

Perceptual hiding of leaking nodes is one advantage of the mixed-initiative graph building approach (discussed in Section 3.3.1). However, while masked at a presentation level, the leaks will still exist in the underlying raw data. Additionally, even with access only to the graph visualization, a malicious attacker might still potentially uncover the leaks by meticulously inspecting and recording the relationships between nodes, edges, or cells that are of interest. While visual clutter in graphs is especially helpful, in an adjacency matrix there is no corresponding form of "cover," which makes heuristic approaches easier to apply (such as computer vision algorithms) for the purpose of reconstructing the underlying data from the visualization.

4.4 Strategies for Preservation Operations

We now discuss recommendations for applying privacy preserving actions to leaking sets of nodes and/or edges. These can help a graph builder to decide which type of graph modification operation should be taken, based on the leak type and what kind of nodes and edges are involved.

As mentioned previously, node and edge deletions result in information loss and should be used with caution. We do not consider this type of action as a viable strategy unless the node or edge can safely be deleted because *a priori* it does not add value to the chart (thus it can be deleted anyways). Succinctly put, we adopt two strategies for graph builders: Node merging is more suitable for addressing *k-anonymity* leaks, while edge bundling is better for *l-diversity* leaks.

Thus, the first step of choosing a privacy action is to identify the type of leak, done by applying the definitions in Section 4.1.1. Next, the user should determine a set of nodes and edges that will be affected by the privacy action. As a node or edge may be involved in multiple privacy issues, resolving one issue can potentially fix others. Our strategy for selecting the nodes and edges to be involved in the privacy action is to maximize the number of issues can be solved via a single operation.

For example, if a privacy leaking node is violating *k-anonymity*, we first collect all other nodes in the graph having the same ontology type as the offending node. Among these nodes, we choose the one with the highest number of *k-anonymity* privacy issues to be merged with the originally selected privacy leaking node. If multiple nodes having the same number of *k-anonymity* leaks, we default to the one with the smallest degree. Although merging the offending node(s) with any other nodes in the graph solves the *k-anonymity* leak, by defaulting the merging to use the most egregiously leaking node, more overall leaks in the graph are resolved.

For an *l-diversity* leak, we retrieve all other edges that share the common ontology types as the privacy leaking edges (*i.e.*, their endpoint nodes are the same types). These edges are grouped according to the ontology nodes that they link to. We count the number of *l-diversity* leaks that the edges violate at the group-level and choose the group with the most *l-diversity* leaks. Excluding any edges that do not have their own *l-diversity* leaks, the remaining edges are bundled with the privacy leaking edges. If multiple groups have the same number of *l-diversity* leaks, we default to the group with the fewest edges in it.

Our prototype system uses these schemes to recommend graph modification actions for leak fixing, but we allow for flexibility and leave the final decision-making to the user. Depending on the con-

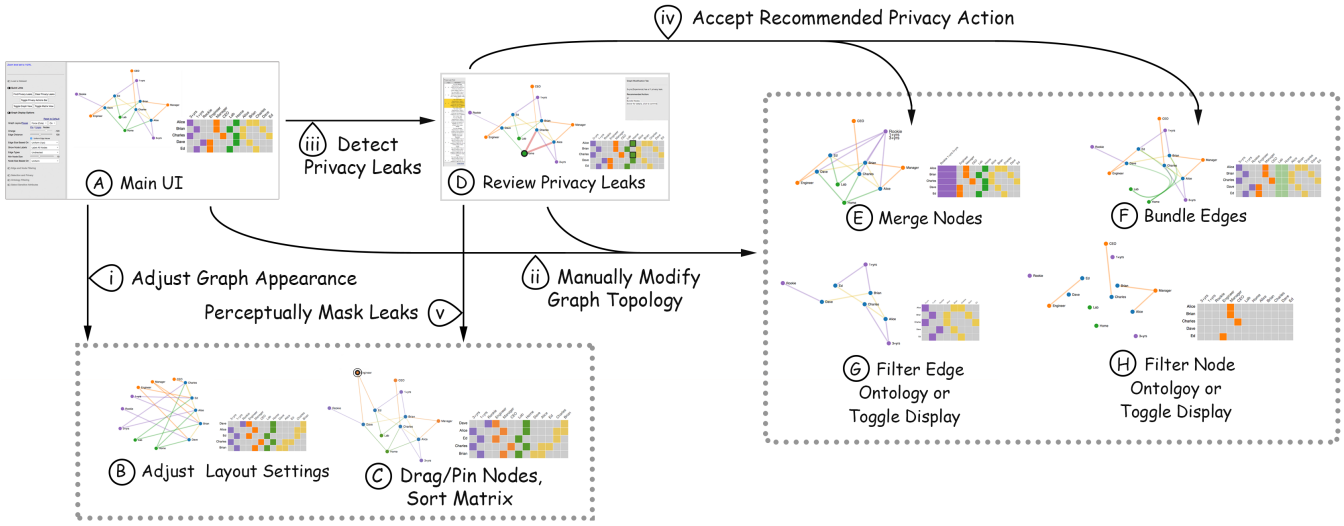


Figure 5: (A) Our system displays the loaded dataset and allows a number of interactions, such as (B-C) modifying graph display settings and (E-H) modifying the graph topology itself. (D) To resolve privacy issues, the user can review a list of detected leaks and choose a desired course of action.

text and the information semantics, a graph builder might have different considerations on how a privacy issue should be handled. For example, a different action than the default recommendation may be chosen to fix a leak (merging a different set of nodes, etc.), or the leak can instead be perceptually masked by updating the graph layout.

5 Prototype System

We have implemented a prototype system for building ontological social network visualizations and performing privacy preservation actions. The system includes graph and matrix viewers with a number of system actions as shown in Figure 5. To create a privacy preserving visualization that retains its semantic utility, we follow an iterative three-step workflow: (1) identify and review privacy issues (Figure 5(iii)), (2) address privacy leaks as desired (Figure 5(ii), (iv), and (v)), and (3) refine the look of the visualization (Figure 5(i) and (ii)).

(A) Upon opening, the user first loads a dataset. The dataset is viewable with both a node-link diagram and an adjacency matrix, though their display can be toggled. Using sidebar controls, a number of graph operations can be performed. (i) To stylistically update the graph, (B) the user can edit the layout constraints and (C) manually position nodes (and sort the matrix). (ii) If the user wants to manually modify the graph's topology, s/he can (E) merge a set of nodes, (F) bundle a set of edges, (G) filter edges by ontology type (or individually toggle an edge's display), and (H) filter nodes by ontology type (or individually toggle a node's display). Using these actions in conjunction with each other, a user can build a graph to semantically emphasize certain aspects of the data. For example, in Figure 2, the user starts out by filtering out all but two types of ontology nodes. S/he then positions these nodes and pins them to the display, and iteratively adds new ontologies until the full dataset is shown.

To handle privacy leaks, the user (iii) invokes the "Detect Privacy Leaks" action. This examines the graph's topology and ontology information and generates a list of privacy leaking nodes and edges. (D) The user can review this list, hovering over leaks with the mouse. Doing so highlights the offending nodes and edges in both the graph and the matrix. Clicking a leak toggles the "Recommended Action" tab, which suggests a graph modification action (either node merging or edge bundling) depending on the context of the leak. To resolve the leak, the user can (iv) accept this recommended action, (ii) manually modify the graph in some other way, or (v) modify the graph's layout and appearance to perceptually mask the leak. (The leak can also be ignored, if deemed unimportant.) If the graph's topology is updated, the list of privacy leaks is refreshed (removing the fixed leak, and possibly others resolved by the fix). If certain ontologies or sets of nodes can safely be ignored (*i.e.*, it does not matter if they are exposed or causing leaks), they can be toggled as unsensitive in the sidebar, and will not show up in the privacy leaks list. The user has the option to continue reviewing and fixing other leaks as well as improving the semantics presented by the visualization(s) until all the important privacy leaks have been fixed and s/he is satisfied with the look of the to-be-presented visualization(s).

6 Case Studies

We demonstrate two case studies with our prototype system showing how ontological graphs can be modified to preserve privacy while still maintaining utility.

6.1 MIT Reality Mining Dataset

The MIT Reality Mining dataset [9] shows the communication, proximity, location, title and activity information from 100 subjects at MIT over the course of the 2004-2005 academic year. From this, we extract activities that happened during the month of April.

Figure 4(a) shows a built node-link diagram of the data. Its node layout has been arranged to emphasize different ontological groupings of people. For example, the green MIT and Boston nodes are popular [Locations] to live. The purple Friends and Restaurant_or_bar [Hangout] places are popular for all groups. The cluster of [Persons] that have Student [Titles] are most sociable with persons who are either New Grad or MediaLab, while Sloan and Senior Grad do not commonly associate with other groups.

Unfortunately for the graph in Figure 4(a), an attacker with sufficient background knowledge could dig out privacy information by carefully examining the view. In Figure 4(d), we highlight two privacy leaks. The first is that there is only one person whose [Title] is Junior Grad and lives at the [Location] Harvard. If an attacker knows this information, s/he will know this person goes to the Friends and Gym [Hangout] places. The second leak is that if we know that someone's [Title] is Senior Grad and lives in MIT, s/he goes to the Friends [Hangout].

Figure 6(a) shows an updated visualization with these (and a number of other) privacy issues fixed. For example, the nodes Sloan and Junior Grad have been merged to address the first privacy issue. Edge bundling has been applied to correct the second leak. Despite a number of actions being applied that modify the graph's appearance and topology, the view is still able to communicate the major information themes that are conveyed in Figure 4(a).

Figure 4(b) shows the same data as Figure 4(a) but with a different layout of the nodes to emphasize a different pattern in the data. For example, this alternate layout highlights that people who live in Boston and MIT have many more social interactions than those living in other places. Additionally, Friends is a common [Hangout] place for people living at MIT, while Gym is popular for those who live in Boston and Central. Because the underlying data presented here is the same as Figure 4(a), privacy detection will reveal the same set of privacy issues.

These leaks can be perceptually masked by adjusting the visual layout and display appearance of the graph. In Figure 6(b), the leaking Senior Grads node (in conjunction with the MIT node, mentioned above) is positioned in such a way that visual clutter makes it difficult for a viewer to detect, though it still exists at a data-level.

In Figure 4(b), the privacy issue that exposes the hangout place of Senior Grads who live in MIT cannot be discerned from the visualization. As a result, we choose not to address this particular privacy issue.

6.2 School Kids Friendship Dataset

The second case study uses a dataset [11] collected from a group of public school students (8th to 12th graders), of which we extract an anonymous subset of the dataset (due to privacy concerns). Figure 7(a) shows a node-link diagram arranged to present ontological semantics for the data. Here, the students that are Asian and Female tend to befriend other females rather than males, as opposed to White Female students, who make friends equally regardless of gender. In datasets like this, attributes such as sexual orientation might be considered a sensitive type of information (that entities would want kept private). By looking at the graph in Figure 7(a), there are several privacy leaks that might expose a student's orientation. For example, there is only one Latino student who lives with Mom Only; his sex orientation is Gay. Similarly, there is only one Female Asian student who lives with Mom Only; her orientation is Bisexual.

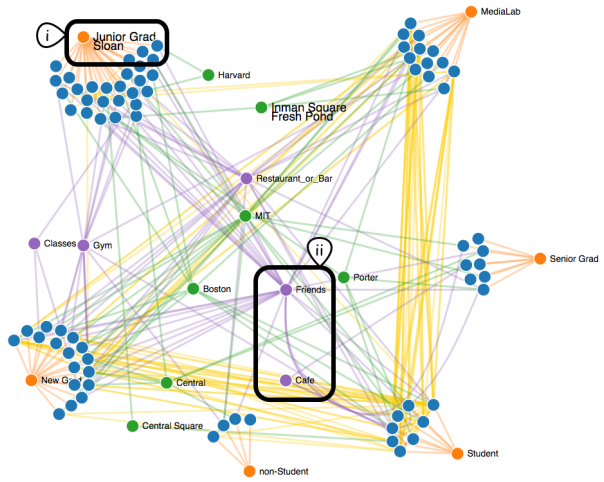
Figure 7(b) shows the visualization where these privacy leaks are addressed. We fix the first leak by merging the Gay and OrientationOther nodes. To further improve the privacy we also bundle the [Sex Orientation] edges that link to both Latino and Male student nodes. Now, a person viewing the graph cannot explicitly discern which Latino student maps to the Gay ontology node. For the second leak, we bundle all [Sex Orientation] edges that link to Female Asian students. Now a viewer is unable to tell which Asian student is Bisexual. Despite modifying the graph's topology to resolve the leaks, the utility of the graph remains high and the overall information presentation is nearly identical to the graph in Figure 7(a). Though the modifications have a small visual effect on the chart, these particular privacy leaks are now handled, ensuring these students do not have their identities violated.

7 Discussion

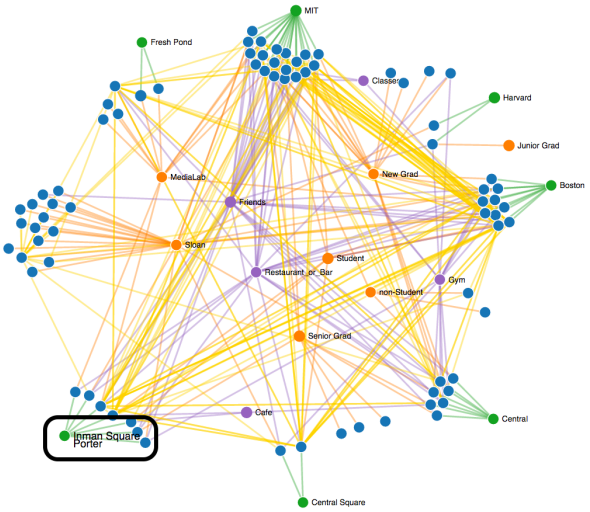
To further assess our prototype system and the techniques discussed in this paper, we conducted a series of interviews with sociology researchers. We also consider areas that require further discussion and can potentially be avenues for future research.

7.1 Feedback from the Sociology Community

Here we summarize feedback and quotations from interviews with four sociologists– two professors, one research fellow, and one



(a) (i) Merging **Junior Grad** and **Sloan** solves a *k-anonymity* leak (k=2). (ii) Edge bundling the **Friends** and **Cafe** nodes solves two *l-diversity* leaks (where l=2).



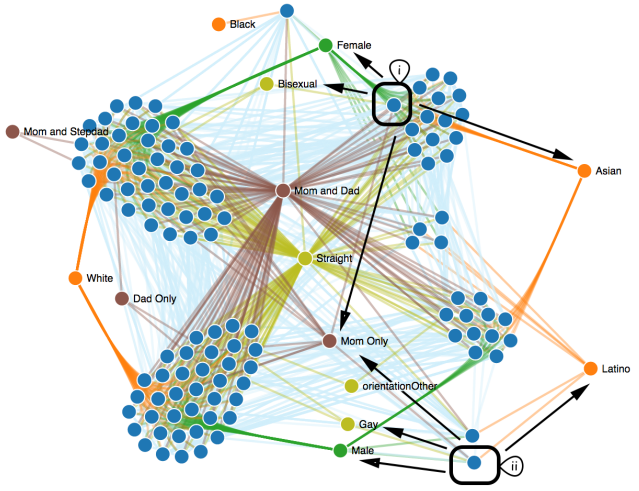
(b) In some instances, a single action can solve multiple types of leaks. Merging **Inman Square** and **Porter** solves both a *k-anonymity* leak (k=2) and an *l-diversity* leak (l=2).

Figure 6: MIT dataset graphs, with privacy operations applied (pre-privacy action graphs are shown in Figure 4(a)-(b)).

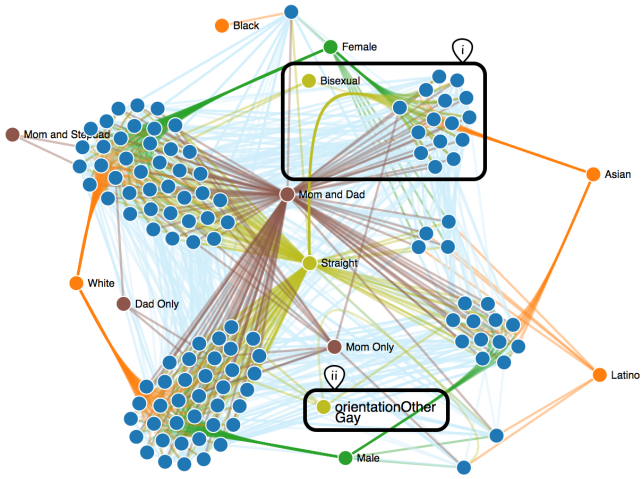
postdoctoral researcher. Each has at least seven years experience in conducting research and works with ontological social networks with sizes of less than 50 to over 500,000 persons (normal ranges usually scale to within hundreds of persons). Overall feedback was positive; we believe that privacy detection algorithms and privacy preserving actions like the ones used in our system can augment current domain efforts, especially in certain data contexts.

In the sociology research community, "*privacy protection is hugely important... the problem of deductive disclosure is one most people are aware of.*" There are several strategies used to ensure privacy of individuals; however each has its own concerns and limitations. Hairballing, a naive form of the perceptual masking from Section 4.3, has long been used: *"historically, the solution to privacy in networks is the hairball problem. [The assumption is] it'd be pretty difficult for someone to identify a person in the graph."* Hiding the origin location of a dataset is also very common (*"our current fortress"*), though this carries the risk of the data later being inadvertently exposed: *"with secondary data though, you might have someone just give away the name."*

For visualization-specific solutions, one technique is visualizing



(a) (i) The only Asian Female who lives with Mom Only has her sex orientation (Bisexual) exposed (an *l-diversity* leak where l=2). (ii) Similarly, the only Latino Male who lives with Mom Only also has his sex orientation (Gay) exposed (a *k-anonymity* leak where k=2).



(b) (i) Bundling all sexual orientation edges going to Asian Females fixes the first leak. (ii) Node merging the Gay and orientationOther nodes fixes the latter.

Figure 7: Friendship dataset graphs, (a) with leaks exposed and (b) then fixed.

only attributes that are deemed insensitive, though this limits what can actually be shown to viewers. Randomness in the form of node (or edge) insertion, deletion, or modification is also used. Uncertainty preserves the privacy, but it introduces error into the visualization and is particularly unfeasible for smaller networks that contain edge cases and outliers. One researcher noted instances where her datasets heavily skewed male: "*adding a single node, especially if it's a woman, can throw off the weight of the network. Introducing too much noise can take away the validity of our results.*" This forces the raw data to be kept unpublished and only statistical and aggregate metrics can be presented. This also precludes visualization of the raw data itself.

In giving feedback, all researchers were surprised by the amount of privacy issues that could actually be detected by our system. Though they knew privacy issues were present in the data, but they did not know how they were revealed at an algorithmic level, nor how to deal with individual cases of privacy leakage, and liked that our system gave both explanations and solutions for issues. One noted, "*the system really helps me examine the potential privacy issues in a dataset. It can save me a lot of effort.*" Three researchers noted targeted privacy preservation actions could be leveraged in the context of the absence of IRBs, which are often used to allow for deductive disclosure of individuals. "*You can not always get the consent of incidentals [especially in egocentric networks],*" so privacy leaks regarding these individuals can be identified and fixed. Another suggested use case was when the location of a dataset cannot be hidden (organizations like workplaces and corporate environments were suggested) or "*jiggered*" by introducing dummy data, "*I like the elegance of the solution. There's no loss of accuracy.*" It was in situations like these especially that our interviewees felt these techniques could be leveraged.

7.2 Future Considerations for Privacy Preservation in Ontological Graph Visualization

As Section 4.1 notes, there is no standard way to define what constitutes a privacy leak. We leverage privacy definitions used in relational data models, but other graph properties can be used to detect leaks (even for non-ontological datasets): node degree [35, 36], edge weight [18], whether a node has a link to dummy nodes intentionally created for the graph in advance by an attacker [2], and embedded attribute information of a node [30] are all examples.

That said, once detection of privacy leaks in a graph is formally defined, preservation can be accomplished using the methods presented in this paper. Moreover, interactive graph building systems (such as our prototype) can provide the benefit of an iterative, mixed-initiative layout. This allows users to focus on the overall readability of the visualization as leaks are fixed.

Unfortunately, the major downside of modifying a graph for privacy preservation is loss of utility. Most data-centric approaches measure utility by examining how well the topological properties of a graph are maintained after anonymization [29, 37]. Common metrics for this include shortest path, node degree distribution, spanning tree topology, and centrality. Integrating these into privacy preservation models is not explored in this paper, though they do pose interesting questions for future work. For example, you can rank anonymization actions for a given graph by how much change they introduce to those graph properties.

As a last note, in this paper we consider the purpose of created graph and adjacency matrix visualizations to be for communication and presentation of dataset results. In many cases, only the created graphic is shared with others, not the raw dataset values. However, at other times it may be desirable or necessary to share the underlying dataset. Since privacy preservation actions such as node merging or edge bundling have an effect on the topology of both graphs and adjacency matrices, an exported raw dataset can reflect these actions.

The term for this type of action is *generalization*, or alternatively, *suppression* [12]. For example, node merging is generalized by using a single row/column to represent the merged node. Edge bundling can be generalized by manipulating edge weight values or annotating additional labels to denote terminal uncertainty for a set of edges. Perceptual masking cannot be suppressed in raw data, so care must be taken in this instance.

8 Conclusion

In this paper, we discuss visualization and privacy preservation of ontological social network data using node-link diagrams and adjacency matrices. We leverage relational data anonymization models to identify potential data-level privacy leaks that can be exposed through visualization. Once identified, leaks can be dealt with through chart modification operations or by adjusting the layout and display, which cognitively influences the way a leak is perceived and can effectively mask the leak from viewer perception.

To transform a privacy leaking chart into one that preserves the security of individuals while still being useful for presentation, we

define an iterative workflow that allows a graph builder to (1) find and examine privacy issues, (2) obfuscate leaks as desired, and (3) refine and stylize the look of the visualization. Using a prototype system, we demonstrate with case studies that show how our approach can create presentable, sharable, and customized visualizations to maintain a tradeoff between privacy considerations and overall chart utility.

ACKNOWLEDGEMENTS

This research is supported in part by the UC Davis RISE program, U.S. National Science Foundation via grants NSF IIS-1528203 and NSF IIS-1320229, and the U.S. Department of Energy through grant DE-FC02-12ER26072.

REFERENCES

[1] B. Bach, E. Pietriga, I. Liccardi, and G. Legostaev. Ontotrix: A hybrid visualization for populated ontologies. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 177–180, New York, NY, USA, 2011. ACM.

[2] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[3] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716, 2016.

[4] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class-based graph anonymization for social network data. *Proc. VLDB Endow.*, 2(1):766–777, Aug. 2009.

[5] J.-K. Chou, Y. Wang, and K.-L. Ma. Privacy preserving event sequence data visualization using a sankey diagram-like representation. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, SA '16, pages 1:1–1:8, New York, NY, USA, 2016. ACM.

[6] S. Das, Ö. Eğecioğlu, and A. E. Abbadi. Anonymizing weighted social network graphs. In *2010 IEEE 26th International Conference on Data Engineering*, pages 904–907, March 2010.

[7] A. Dasgupta and R. Kosara. Adaptive privacy-preserving visualization using parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2241–2248, 2011.

[8] A. Dasgupta, E. Maguire, A. Abdul-Rahman, and M. Chen. Opportunities and challenges for privacy-preserving visualization of electronic health record data. In *IEEE VIS 2014 Workshop on Visualization of Electronic Health Records*, 2014.

[9] N. Eagle and A. (Sandy) Pentland. Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, Mar. 2006.

[10] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, May 2010.

[11] R. Faris and D. Felmlee. Social networks and aggression at the wheatley school. Report for CNN, available at `http://i2.cdn.turner.com/cnn/2011/images/10/10/findings.from.the.wheatley.school.pdf`, 2012.

[12] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42(4):14:1–14:53, June 2010.

[13] M. Hay, G. Miklau, D. Jensen, D. Towsley, and C. Li. Resisting structural re-identification in anonymized social networks. *The VLDB Journal*, 19(6):797–823, 2010.

[14] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. Technical report, University of Massachusetts Amherst, 2007.

[15] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, Sept. 2006.

[16] A. Hussain, K. Latif, A. T. Rextin, A. Hayat, and M. Alam. Scalable visualization of semantic nets using power-law graphs. *Applied Mathematics & Information Sciences*, 8(1):355, 2014.

[17] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 93–106, New York, NY, USA, 2008. ACM.

[18] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preservation in social networks with sensitive edge weights. In *Proceedings of the ninth SIAM international conference on data mining*, pages 954–965, 2009.

[19] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007.

[20] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *2009 30th IEEE Symposium on Security and Privacy*, pages 173–187, May 2009.

[21] T. Oellinger and P. O. Wennerberg. Ontology based modeling and visualization of social networks for the web. *GI Jahrestagung*, 2(94):489–497, 2007.

[22] Z. Shen and K. L. Ma. Mobivis: A visualization system for exploring mobile data. In *2008 IEEE Pacific Visualization Symposium*, pages 175–182, March 2008.

[23] Z. Shen, K.-L. Ma, and T. Eliassi-Rad. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1427–1439, Nov. 2006.

[24] L. Shi, Q. Liao, H. Tong, Y. Hu, Y. Zhao, and C. Lin. Hierarchical focus+ context heterogeneous network visualization. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 89–96. IEEE, 2014.

[25] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002.

[26] M. Wattenberg. Visual exploration of multivariate graphs. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 811–819. ACM, 2006.

[27] P. Wu and S. Li. Social network visualization via domain ontology. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–4, Dec 2009.

[28] P. Wu and S. Li. Social network analysis layout algorithm under ontology model. *Journal of Software*, 6(7):1321–1328, 2011.

[29] X. Wu, X. Ying, K. Liu, and L. Chen. *Managing and Mining Graph Data*, chapter A Survey of Privacy-Preservation of Graphs and Social Networks, pages 421–453. Springer US, Boston, MA, 2010.

[30] M. Yuan, L. Chen, P. S. Yu, and T. Yu. Protecting sensitive labels in social network data anonymization. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):633–647, March 2013.

[31] H. Zakerzadeh, C. C. Aggarwal, and K. Barker. Big graph privacy. In *8th International Workshop on Privacy and Anonymity in the Information Society (PAIS)*, 2015.

[32] L. Zhang and W. Zhang. Edge anonymity in social network graphs. In *CSE*, 2009.

[33] J. Zhao, M. Glueck, S. Breslav, F. Chevalier, and A. Khan. Annotation graphs: A graph-based visualization for meta-analysis of data based on user-authored annotations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):261–270, Jan 2017.

[34] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *Proceedings of the 1st ACM SIGKDD International Conference on Privacy, Security, and Trust in KDD*, PinKDD'07, pages 153–171, Berlin, Heidelberg, 2008. Springer-Verlag.

[35] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 506–515, Washington, DC, USA, 2008. IEEE Computer Society.

[36] B. Zhou and J. Pei. The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowl. Inf. Syst.*, 28(1):47–77, July 2011.

[37] B. Zhou, J. Pei, and W. Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explor. Newsl.*, 10(2):12–22, Dec. 2008.