An Empirical Study on Perceptually Masking Privacy in Graph Visualizations

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

University of California, Davis

Abstract

Researchers such as sociologists create visualizations of multivariate node-link diagrams to present findings about the relationships in communities. Unfortunately, such visualizations can inadvertently expose the ostensibly private identities of the persons that make up the dataset. By purposely violating graph readability metrics for a small region of the graph, we conjecture that local, exposed privacy leaks may be perceptually masked from easy recognition. In particular, we consider three commonly known metrics—edge crossing, node clustering, and node-edge overlapping—as a strategy to hide leaks. We evaluate the effectiveness of violating these metrics by conducting a user study that measures subject performance at visually searching for and identifying a privacy leak. Results show that when more masking operations are applied, participants needed more time to locate the privacy leak, though exhaustive, brute force search can eventually find it. We suggest future directions on how perceptual masking can be a viable strategy, primarily where modifying the underlying network structure is unfeasible.

Index Terms: Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

In fields such as sociology, economics, and marketing, researchers study social networks to investigate populations of interest. Visualization with node-link diagrams is an effective way to display large-scale community structures as well as the specific relationships between individual persons.

To enrich the overall utility—i.e. the analytic value—of these visualizations, additional personal and semantic information can be encoded into the graphs [17,20,21]. Figure 1(a) shows one common approach: appending nodes that contain data attribute values. This is known as an *ontology graph*: a node-link diagram of a social dataset composed of heterogeneous nodes that represent either persons in the network or the attribute values that a person can have.

Visualizations such as these are often found in scientific publications and technical presentations. Publicly presenting a network in this way, even when only showing a paper figure or powerpoint illustration, carries an inherent risk of *privacy exposure*. This means that the visualization can inadvertently reveal enough topological or relational information that specific individuals are identifiable, even when the raw dataset is withheld. For example, by examining the visualization, one may be able to find a group of nodes (persons) sharing a single sensitive property value that also share the same set of non-sensitive, publicly-known property values. In such a case, we say the privacy of this set of individuals is leaked (see Section 3.2 for more explanation). We refer to these persons with exposed privacy as *privacy exposed nodes* (PENs).

A straightforward solution to this problem is to modify (or simplify) the topology of the dataset to remove any and all PENs prior to creating the visualization. This approach has been widely studied in the data mining community (e.g. [27]); common operations include adding nodes/edges, merging nodes, deleting node/edges, swapping edges, etc. Unfortunately, such a broad-based approach is not always feasible. Simplifying a dataset's topology can dramatically change its resultant utility. As stated by a sociologist, *"introducing too much noise can take away the validity of our results"* [5]. This is a critical factor when considering the extent of privacy preservation that can be achieved and the number of privacy leaks that will be present in the resultant visualization.

Recently, a technique called *perceptual masking* was proposed for privacy preservation in ontology graphs [5]. The idea is to adjust the graph's layout to hide PENs with graph clutter. The advantage is the topology remains unmodified. However, this leads to an interesting question and follow-up, left unexplored in [5]: *Does perceptual masking actually work? If so, to what extent?*

This question motivates the study reported in this paper: an initial evaluation of the effectiveness of perceptual masking in ontology graph visualizations. Normally, layout algorithms try to optimize various aesthetic-based criteria to produce high overall readability. We posit that cluttering a small region of the graph—violating the layout aesthetics in localized area(s)—could be a successful masking strategy that makes it harder (or potentially impossible) to visually identify PENs from the visualization.

In particular, we consider violating three aesthetic criteria: (1) increasing the number of edge crossings, (2) increasing the amount of edge-node overlapping, and (3) spatially clustering PENs with non-leaking nodes. We test combinations of these strategies in a controlled user study. Specifically, the study is designed to answer the following research question: *How does the application of perceptual masking affect a person's ability to identify PENs in an ontology graph?* That is, can we still find any nodes that leak privacy? If so, how effective are the different cluttering strategies?

For stimuli, we synthesize a set of ontology graphs that each contains a single set of PENs. The layouts of these graphs follow common aesthetic criteria except for regions containing the PENs. These are obfuscated using a combination of the three masking strategies. Study subjects are tasked to visually scan the graph to identify the PENs.

Our results indicate that PENs (for our stimuli) can eventually be found via exhaustive serial scanning. Despite this, different combinations of masking strategies make it easier or harder (measured by task completion time) to do so, indicating that effective perceptual masking can delay visual identification of privacy leaks. This is notable because in instances where the graph is only temporarily shown, small changes to graph layout might be enough to ensure privacy is preserved.

While we consider this study to be initial and exploratory, the findings are useful for practitioners who need to "weakly hide" privacy leaks in their dataset visualizations. To promote future research directions, we summarize and reflect on participant feedback, as

*e-mail: jkchou@ucdavis.edu

†e-mail:cjbryan@ucdavis.edu

‡e-mail: jinli@ucdavis.edu

§e-mail: ma@cs.ucdavis.edu

978-1-5386-8194-7/18/$31.00 ©2018 IEEE

Metric Type		Metric Name and Description
Node	(1)	**Node clustering**. If two nodes have semantic commonality, such as link to a similar set of other nodes, then they should be placed close to each other.
	(2)	**Node separation**. Distance between any two nodes should be bigger than a certain value so that they do not occlude each other and can be easily distinguished.
	(3)	**Node-edge separation**. Nodes should be kept from coming too close to edges, or vice versa.
	(4)	**Maximize node orthogonality**. Enforcing the placement of nodes into an imaginary 2D grid.
Edge	(5)	**Minimize edge crossings**. Reduce the number of pairs of edges which geometrically intersect each other.
	(6)	**Maximize edge crossing angles**. Maximize the average cosine angles of crossing edges.
	(7)	**Make edge lengths uniform**. Keep the variability of edge lengths to be small.
	(8)	**Minimize edge length**. Minimizing both the sum of all edge lengths and the maximum edge length.
	(9)	**Minimize edge bends**. Minimize the number and the angle of edge bends within a graph (especially for path finding tasks).
	(10)	**Maximize minimum edge angles** Maximize the angle between any two neighboring edges linking to the same node.
	(11)	**Maximize edge orthogonality** Similar to node orthogonality.
Layout	(12)	**Consistent flow direction**. The direction that edges pointed to, such as upwards or downwards, should be as consistent as possible (for directed graph only).
	(13)	**Aspect ratio**. Keep the shape (aspect ratio) of the graph to be close to the aspect raio of the display.
	(14)	**Reflect inherent symmetry**. The extent to which the layout is locally (a subgraph of the graph) or globally (the whole graph) symmetrical in three directions: vertically, horizontally, and diagonally.
	(15)	**Conform to the frame**. The nodes fill the the available drawing space without going outside of its boundaries.

Table 1: Common aesthetic criteria for measuring graph readability and quality.

well as how perceptual masking strategies can be applied to complex and scalable graph datasets for in-depth evaluation.

2 Background and Related Work

Graph Drawing and Aesthetic Criteria. In mathematics and computer science, graph drawing is concerned with designing algorithms for visualizing and laying out graphs [2]. Although there is no "objectively best" layout for a graph, various aesthetic properties are considered good proxies for evaluating quality and readability. Force-directed methods, as an example, are popular layout techniques. Nodes and edges are treated as a system of interacting physical objects that push and pull on each other. When the system reaches a state of equilibrium, the resulting layout is, almost serendipitously, organized in a way that is perceptually and aesthetically pleasing [10].

Table 1 lists common aesthetic-based metrics for graphs, based on a survey by Bennett et al. [3] and several other works on graph readability [1, 10, 23–25]. It is important to note that not all graph aesthetics can be simultaneously optimized, since they sometimes conflict with each other [2]. For example, minimizing edge lengths often introduces edge crossings. Some graph types can ignore certain metrics; un-directed graphs with straight edges do not have to worry about (9) edge bends and (12) edge directions.

In general, popular layout algorithms are designed to optimize at least a subset of these metrics. In our case, the graphs used in the study have overall high readability, except for a small region where we intentionally violate aesthetic criteria to hide privacy leaks.

Privacy Concerns for Social Network Data By analyzing the topology of social network datasets, privacy leaks can be identified by mining features like node degree, neighborhood information, shortest paths, edge weights, and entity groupings [4, 7, 14, 27, 28]. When additional attributes are visualized (i.e., relational data about nodes or edges, as with ontology graphs), syntactic privacy models can be employed. The two most common syntactic models, *k-anonymity* [22] and *l-diversity* [15], are used by Chou et al. [5] to identify leaks in sociology graph visualizations, while these types of techniques are largely applied for tabular datasets. To identify privacy leaks, sets of nodes that form equivalence classes are examined to see if they violate a minimum *anonymity* and/or *diversity* criteria. Common techniques for fixing a privacy leak in a graph modify the graph's topology: merging nodes, deleting nodes/edge, and bundling edges.

Randomization strategies are also employed to preserve privacy, including adding "dummy" nodes and edges [13] and swapping edges between nodes [26]. Unfortunately, randomization introduces uncertainty and error into the dataset. A simpler option is hiding the source of the dataset or only publishing a few attributes that are collected. This is mentioned by the sociologists as a common practice in [5], but is difficult to successfully implement in practice.

Perceptual masking, introduced in [5], was proposed as a topology-preserving tactic, especially for "lower risk" (less serious) privacy leaks. Since it does not modify the graph at a data-level, it goes against many of the aforementioned approaches. In this paper, we conduct an initial quantification of its efficacy.

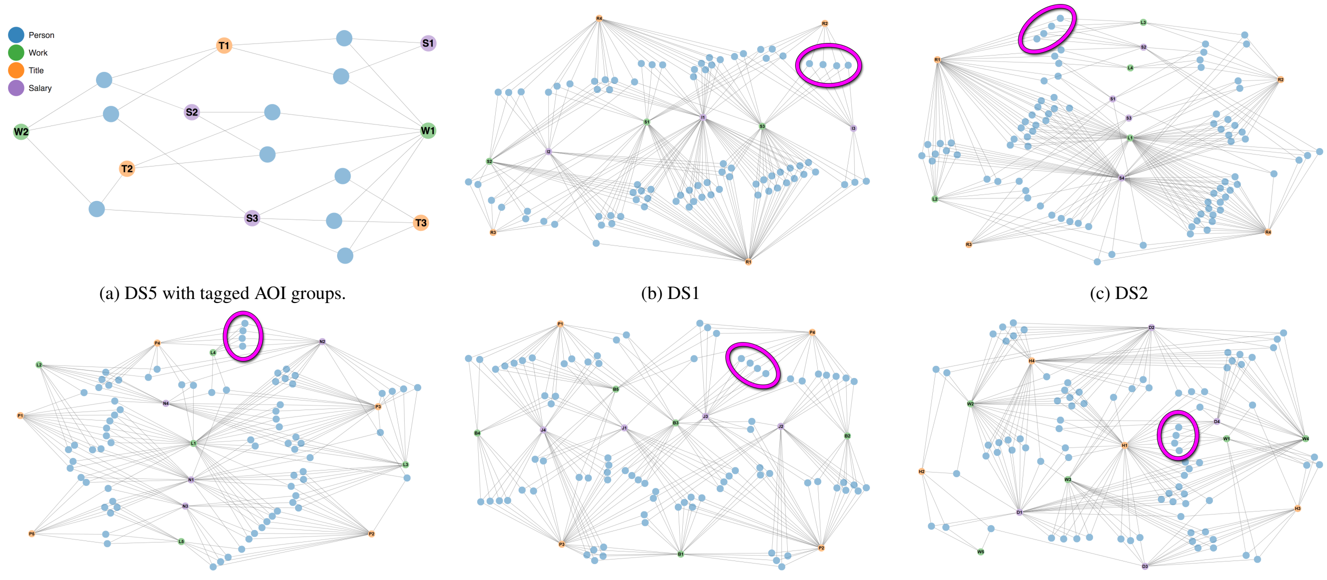
Perceptual Studies in Visualization At a high level, *graphical perception* is concerned with how we cognitively perceive visualizations [6]. At the highest level, the choice of technique affects how we interpret the data, such as using bar versus line charts [19]. Marks and other graphical encoding choices further affect our perceptual accuracy [6].

Healey and Enns discuss several relevant perceptual concerns for visualizations, especially regarding attention, preattentive processing (i.e., pop-out), and visual search [12]. As regards our study, we design the stimuli to avoid introducing pop-out features (as defined by Healey and Enns—unique colors, shapes, sizes, etc.). This means that when subjects search for privacy leaking nodes, they will probably have to serially scan through the graph. This allows us to better assess the efficacy of different masking strategies.

3 Ontology Graphs: Visualization and Privacy Concerns

In this section, we formally define what constitutes an ontology graph following the notations used in [5, 20, 21]. An ontology graph can be treated as a standard social network augmented with additional attribute nodes (termed ontologies) that present additional information about the individuals in the network.

Let $G = (V,E,vt,et)$ denote a graph and $OG = (T_V,T_E)$ denote its associated ontology information. V and E represent the sets of vertices and edges in the graph, respectively. $T_V = \{t_1,t_2,...,t_m\}$ and $T_E = \{(t_i,t_j) : t_i,t_j \in T_V\}$ are the sets of ontology types that correspond to the vertices and edges. For a vertex $v \in V$, $vt(v)$ refers



(a) DS5 with tagged AOI groups.

(b) DS1

(c) DS2

(d) DS3

(e) DS4

(f) DS5

Figure 1: (a) An exemplar ontology graph showing the Tutorial dataset from Table 2. Blue person nodes connect to purple, green, and orange ontology nodes; purple ontology nodes are considered sensitive. (b)-(f) The datasets of our user study, shown in their default layout without any perceptual masking strategies applied (i.e., the None condition). The set of privacy exposed nodes (PENs) are highlighted for each graph.

to its ontology type. Similarly, $et(e)$ denotes the ontology type of an edge e.

3.1 Ontology Graph Visualization

Figure 1(a) shows an example ontology graph visualization of a small, synthetic social network. Blue nodes represent persons, while other colored nodes represent different types of attributes: green for working location, orange for title, and purple for salary. Edges connecting people to attribute nodes specify personal information. For example, the person placed on the bottom left of Figure 1(a) works at **W2**, has a title of **T2**, and receives a salary of **S3**.

It is important to note that the graphs in our study do not include person-person edges. As adding these edges increases the complexity of the graph, this might also inadvertently introduce unexpected effect of privacy masking or graph cluttering– a confounding variable. Removing these edges from consideration helps us more precisely determine the effects of perceptual masking operations under controlled conditions. We discuss the application to more complicated, real-world datasets in Section 7.

While ontology graphs can be enlivened by adjusting visual channels like normal graphs (including node size scaling, node shapes, edge thickness, directed edges, etc.), for study consistency we style the graphs that adhere to the following rules:

Node shape: Each node is rendered as a circle without a border.

Node size: Node radius is set to 28 px.

Node color: Node opacity is 50%. Nodes for persons are colored blue (hex code #1$f77b4$). Nodes for sensitive attributes (see Sec. 3.2 below) are purple (#9467bd), while non-sensitive nodes are green (#2$ca02c$) and orange (#$ff7f0e$).

Node labels: Blue nodes representing persons are unlabeled. Attribute nodes are labeled as such: The first letter of its ontology type (i.e., if the ontology is Title, then **T**) is appended with a sequential numbering of nodes in the ontology: **T1**, **T2**, **T3**, etc. Label fonts are sans-serif, black in color, and at 17 px size. In this way, they both fit inside nodes and are easily readable.

Edge styling: Edges are rendered as straight line segments between connected nodes. Edge width is set to 1 px with dark gray color (#777777) at 80% opacity. This makes both edges and nodes distinguishable when edge-node overlapping occurs in conditions S2 and S4 (see Section 4.2).

Color key: A legend is placed in the top-left corner of each graph, giving the full name (and color) of each ontology type.

3.2 Defining Privacy Leaks in Ontology Graphs

In the context of ontological social networks, attributes are deemed *sensitive* if revealing the value of those specific attributes is considered an invasion of privacy– for example, a person's salary. In contrast, an attribute is non-sensitive if it is publicly shared knowledge, such as a person's company of employment and job title. In Figure 1(a), the purple ontology nodes represent sensitive salary values; green and orange ontology nodes are non-sensitive attributes. To identify privacy leaks in an ontology graph, we follow the same assumption and utilize two commonly used privacy models, namely *k-anonymity* [22] and *l-diversity* [15], as in [5].

k-anonymity is defined such that an equivalence class must contain at least k records. In an ontology graph, an equivalence class refers to a group of individuals who share an exactly intersecting set of non-sensitive attribute values. In the lower left of Figure 1(a), we can see there is only one person who works at **W2** with job title **T2** (equivalence class $\{$ **W2 + T2** $\}$), linking to *sensitive* salary node **S3**. This is a *k-anonymity* leak if we set $k = 2$.

l-diversity further extends *k-anonymity* as it considers the number of sensitive attributes that the individuals in the same equivalence class must map to. If the number is smaller than l, it is considered a privacy violation [15]. For example, the two persons near the upper right corner of Figure 1(a) belong to the same equivalence class of $\{$ **W1 + T1** $\}$ while both also link to a single salary node: **S1**. This is an *l-diversity* leak if $l = 2$, since this equivalence class maps to less than two sensitive attribute values. This means if we know someone who works at **W1** and has a job title **T1**, we can specifically figure out his salary is **S1**.

If a person is tasked with finding privacy leaks in graphs, these types of equivalence class comparisons can be performed. To keep our study design consistent, we only consider privacy leaks that are *l-diversity* with $l = 2$, which means that study participants must find the equivalence class where the referenced person nodes only link to a single sensitive ontology node. As a shorthard, we define the following terms for use in our study and for the rest of the paper:

PLONs (privacy leaking ontology nodes): The set of ontology attribute nodes causing a privacy leak. The graphs we design for the study only contain one *2-diversity* leak, so each graph contains one set of PLONs: a pair of one green node and one orange node forming an equivalence class, which decides a unique sensitive value represented by one purple node.

PENs (privacy exposed nodes): The set of blue person nodes whose privacy is being exposed. PENs have edges to the PLONs.

OONs (other ontology nodes): Ontology nodes in the graph that do not cause *2-diversity* privacy leaks. OONs do not have edges to PLONs nor PENs.

Person nodes that do not leak privacy are left un-categorized; they can link to OONs and PLONs (though not *only* to PLONs, otherwise they would be PENs).

4 Generating Study Stimuli

We discuss how we synthesize datasets and our layout and visualization considerations. We then describe three privacy masking strategies, while the combinations of which result in five conditions for our user study.

4.1 Dataset Generation and Default Graph Layouts

We create artificial datasets for our study to control for size and complexity. Table 2 lists the statistics for each. The first three datasets are used for training, and the other five are user study graphs (DS1–DS5, Figure 1(b)-(f)). Datasets having more than 90 nodes, referring as "regular-sized datasets" hereafter, were created based on real-world datasets, such as the ones in [8, 9, 16].

We first extracted and derived ontological social networks from those datasets. Then, we hand-crafted the underlying data so they each has three types of attributes and between 10–13 attribute nodes. We further made each regular-sized dataset containing one single privacy leak (i.e., one set of PENs), which is always a *2-diversity* violation involving a single set of PLONs (one orange, one green, one purple). While not identical, the regular-sized datasets have similar topological structures. This is done to minimize the chance of the graphs themselves being a factor that affects the study results. As noted earlier, in the visualization orange and green nodes always represent non-sensitive attributes, while purple nodes are sensitive attributes. In addition, no person-person edges are included.

Similar to [5], we first laid out the graphs of our datasets based on a force-directed algorithm and later manually adjusted to account for ontology semantics. In addition, we tried our best to adhere the following aesthetic-based criteria as much as possible: distances of nodes in clusters should remain uniform, long edges should be avoided, edge angles linking to the same pair of nodes should be made as even as possible, and the graph should look symmetric. Finally, the aspect ratio of the resultant graph is kept close to 1 : 1.6.

4.2 Applying Perceptual Masking to Preserve Privacy

As the ontology graph visualization conveys information about the social network, it also provides the opportunity for a person to visually scan for and identify PENs (if any exist). The set of PENs for DS1–DS5 are highlighted in Figure 1(b)-(f). To hide these leaks from easy identification, we manually adjust only a subset of the graph in a way that violates one or more readability criterion. The intent is to make the set of PENs in each graph harder to notice, or which we call "being perceptually masked".

We consider three strategies to "perceptually mask" the set of PENs in each graph, see Figure 2 for examples. For each, we briefly describe *why* we consider it as an applicable strategy to violate.

Dataset Name	Total Nodes	Ontology Nodes	Total Edges
Tutorial	18	8	30
Practice (small)	39	11	84
Practice (regular)	92	12	240
User Study #1 (DS1)	94	10	252
User Study #2 (DS2)	98	12	258
User Study #3 (DS3)	92	13	237
User Study #4 (DS4)	98	13	255
User Study #5 (DS5)	98	13	255

Table 2: Datasets used in the study.

Increase the number of edge crossings. Reducing edge crossings is one of the most agreed-upon criteria for improving graph readability [1, 3, 11, 18, 23]. Conversely, increasing edge crossings is expected to have a counter effect. Our strategy is to re-position the PENs with a small set of non-privacy leaking nodes such that the edges of these two groups intersect with each other, see Figure 2(b).

Increase the amount of edge-node overlapping. Having edges cross through (or over) nodes is a major factor in making it difficult to find common neighboring nodes [11]. We purposefully place PENs such that they are overlapped by edges from non-privacy leaking nodes, see Figure 2(c) and 2(e).

Introduce unnecessary node clusters. In an optimal ontology graph layout, persons that share equivalence classes are placed into discrete spatial clusters. By intermingling PEN-persons with other, non-privacy leaking persons in one single cluster, we make it harder to trace edges, see Figure 2(d)-(e).

While additional strategies can certainly be considered, in this paper we initially focus on this set of three primary ones. In addition to keeping the study variables to a reasonable scale, there are other reasons for not considering other strategies. Limiting potential study confounds and ensuring a controlled study is an important consideration. For example, it is difficult to enforce a unified angle (within a small range) for all edge crossings involving the PENs. Edge length is a metric that takes the entire graph into account. We also do not consider minimizing edge angles as a strategy because it would introduce co-linearity, which may have a similar effect as edge bundling (a "data-level" operation introduced in [5]).

Each strategy we considered takes a slightly different perspective in how it perceptually masks privacy leaks. However, all are alike in only increasing visual clutter for a restricted portion of the graph. Some of them are also integral. For example, introducing edge-overlapping or node clustering will inevitably increase the number of edge crossings. Therefore, to create a set of discrete conditions for the study, we use combinations of the three strategies.

None: No perceptual masking is introduced, the default layout.

S1: Only increased edge crossings are introduced.

S2: Both increased edge crossings and increased edge-node overlap are introduced.

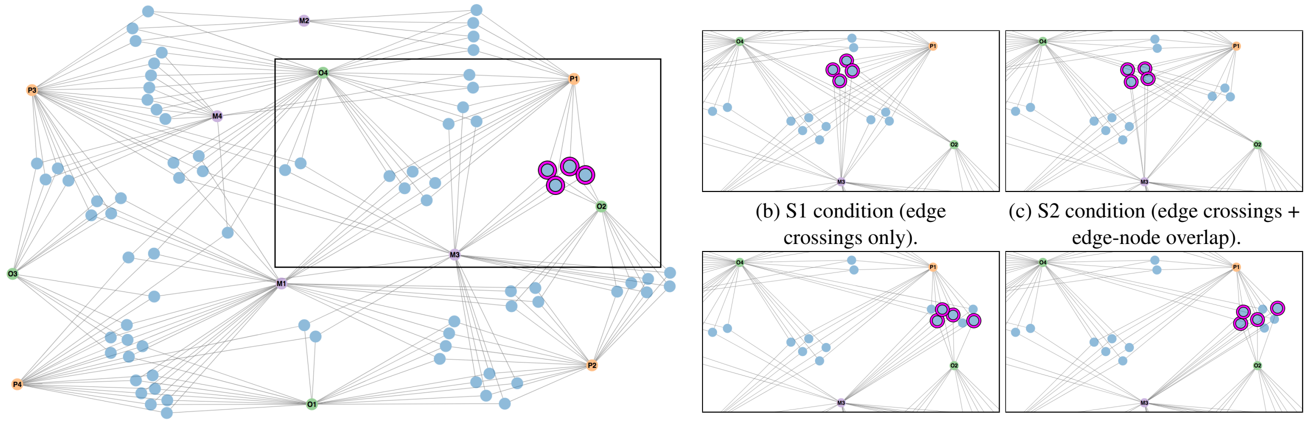
S3: Both increased edge crossings and unnecessary node clustering are introduced.

S4: All three strategies are introduced.

These conditions are labeled in Figure 2. By (manually) changing only a small region of the graph from the default layout, we ensure that overall graph aesthetics and readability remain high.

5 User Study

The study design is within-subject with one factor (the perceptual masking operation) and five conditions (None, S1-S4). Subjects are given one task: when shown a graph, find and click on (with a mouse) the set of PENs. For each trial, we record task completion time and correctness.



(a) None condition (no perceptual masking applied to privacy leaking nodes).

(b) S1 condition (edge crossings only).

(c) S2 condition (edge crossings + edge-node overlap).

(d) S3 condition (edge crossings + node clustering).

(e) S4 condition (edge crossings + edge-node overlap + node clustering).

Figure 2: Showing the five conditions for perceptually masking privacy leaking nodes (PENs) using the Practice (regular) dataset. (a) The None condition is the default layout, optimized for readability. (b)-(e) Perceptual masking strategies only adjust the layout within the selected region. The set of PENs is circled in each graph.

Each participant sees five total graphs, one each of DS1–DS5 (seeing each condition and dataset only once). To counterbalance potential learning effects for finding and identifying leaks, the order with which conditions and datasets are displayed is set using Latin squares. For every five participants, we generate two 5×5 Latin square matrices. The first matrix assigns the order of conditions and the second matrix determines the order of datasets.

5.1 Participants and Environmental Setup

A total of 27 participants (20 males) were recruited, ranging between 21-41 years in age ($\mu = 27.74$, $\sigma = 5.12$). 25 participants had a computer science background; the other two were in design and epidemiology major, respectively. All participants were familiar with and could interpret node-link diagrams; this was considered as prerequisite knowledge for taking the study (as like one would have to know how to read a graph in order to identify any PENs it contained). 9 participants reported having normal vision, 14 wore glasses, 3 wore contact lens, and 1 participant had previously received Lasik surgery but required no current vision correction. All participants reported normal color vision.

All participants completed the study in a campus research lab, a quiet, office-like environment with artificial lighting. On average, participants took between 20-30 minutes to complete the entire study. The biggest variant in total time was how long the tutorial and training stages took, as these were not time-restricted. Some participants also completed the main tasks under the time limit, and so finished the main study stage faster.

All visualizations and the user interface were shown to the participants using a 20-inch Dell monitor with 1680 × 1050 pixel resolution. Each participant was seated at approximately three-quarters monitor height at a distance of 65-70 cm from the monitor. During the main study, mouse click events and timing data were recorded to a local server as log files.

5.2 Study Procedure

The study procedure was the same for all participants, consisting of three stages: (1) tutorial, (2) training, and (3) the main study.

Tutorial Stage: Participants read through a short slide show featuring the Tutorial dataset (Figure 1(a)). These slides explained the concepts of ontology graphs, equivalence classes, sensitive attributes, and how to visually identify a set of PENs.

Training Stage: Participants performed two training rounds of the study task: identifying a set of PENs in a graph. The two "Practice" datasets described in Table 2 were used. This stage was meant to familiarize participants with the study interface and to give them sufficient exercise in finding privacy leaks.

Two interactions were available within the interface: (1) moving the mouse cursor around the screen, and (2) clicking on a node to toggle it as selected or deselected. While graph visualization systems normally include many other interactions, such as edge selection, node and edge filtering, node dragging, zooming, and panning, these interactions provide computational assistance for identifying (or even directly pointing out) privacy leaks in the graph. Since we focus on visual search, we did not provide these in our study interface.

During the tutorial and training stages, an administrator was present to answer questions and ensure that participants correctly learned what privacy leaks are and how they can be identified. There was no time limit for these stages; participants could ask questions and practice until they felt confident to proceed to the main study stage. To conclude the training stage, the administrator demonstrated the masking conditions (S1 to S4) that participants might encounter in the main study (using the Practice (regular) dataset).

Main Study Stage: Participants first completed a demographics form. They were informed that this stage consisted of five graphs, each containing (like the practice stage) a set of PENs with between 1–5 nodes.

Participants then proceeded through the five graphs. For each graph, they were told there is a time limit of three minutes to complete the task, determined based a pilot study with 3 participants. (We found that participants began losing focus after viewing a specific graph for more than three minutes without rest.) This also allows us to limit total study time to approximately 30 minutes.

If a participant finished a graph before the time limit, s/he could click on a "Time!" button. If participants had not made a selection within three minutes, they were asked to make a guess to complete the task. Between each graph, a break screen allowed a participant to de-stress and prepare for the subsequent graph.

After finishing the study, participants were inquired to provide their thoughts on *if* perceptual masking strategies were effective at hiding privacy leaks and why this was so. A summary of this qualitative feedback is discussed in Section 7.3

Please see the supplemental materials for screenshots of our study interface, user study images at full resolution, and the slides used for the tutorial stage.

5.3 Study Hypotheses

We expected to observe the following results with regards to the effectiveness of the applied perceptual masking techniques:

H1 : As compared to the default layout condition (None), applying only edge crossing (condition S1) will not affect participant performance in identifying PENs (both in time and correctness). That is, None $\approx$ S1.

H2 : When combining more than one perceptual masking strategy (S2–S4), participants will be worse at identifying privacy, both in time and correctness, as compared to None and S1. That is, $\{S2, S3, S4\} > \{None, S1\}$ in masking privacy.

H3 : Of the conditions that combine multiple perceptual masking strategies, S4 (which combines all three) will have the worst performance (both in time and correctness) when compared to the conditions that use two strategies (S2 and S3). That is, $\{S4\} > \{S2, S3\}$ in perceptually masking privacy.

6 User Study Performance Results

Before being able to analyze the study results, we checked the response scores for outliers. One participant failed to correctly identify even one PEN across the whole study (i.e., in every graph, they clicked on 0/4 leaking nodes). In reviewing this participant's response times, we noticed she hit the three-minute mark for all five graphs including None (the default, "easy" layout). Because of this, her results were removed from our analysis.

This leaves data for 26 participants, which we analyze in terms of time spent to locate PENs and correctness at the task. For timing data, because we set a three-minute limit for the participants to perform the task under each condition, the data distribution is heavily left-skewed.

Because the data does not follow normal distribution, we use non-parametric Friedman's tests to evaluate the existence of statistical effects by condition. For post-hoc analysis, we use pairwise Wilcoxon signed-rank test with Holm correction. The significance level (α) is set at 0.05. Figure 3 plots the timing data results; Table 3 displays descriptive statistics of the raw timing values.

6.1 Total Time Spent on Each Graph

We first evaluate how long participants spent on each graph. This timing is determined either by the participant clicking the "Time!" button (denoting they believe they have successfully finished the task) or by reaching the three-minute limit, whichever comes first. Figure 3(a) shows the box plots, while a more detailed set of numeric results is in Table 3 under the "Total Time Spent" columns.

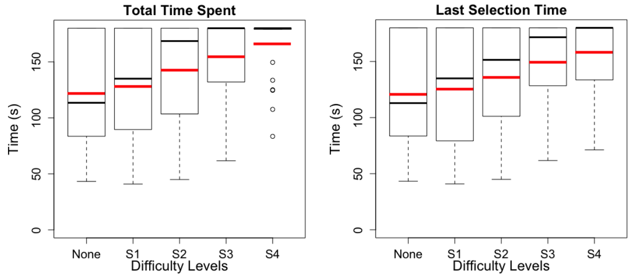
A Friedman's test shows a statistically significant effect between conditions on total time spent in a graph ($\chi^2(4) = 21.136, p = 0.0002976$). This indicates that the different masking conditions affect how long it takes participants to complete their task (when accounting for the time limit cut-off). Post-hoc analysis indicates that None is significantly faster than S3 ($p = 0.034$) and S4 ($p = 0.021$), while S2 ($p = 0.038$) is also significantly faster than S4.

6.2 Time at Last Node Selection

While administering the study, we noticed that after clicking a presumed set of PENs, participants waited different amounts of time before clicking the "Time!" button to finish the task. Some participants were quite cautious: after clicking a set of nodes, they scanned the graph multiple times to review and confirm their answers.

We additionally analyze the conditions based on the time of the last node click. This data is plotted in Figure 3(b) with numeric values shown in Table 3 under the "Last Click Time" section.

The "Last Click Time" data in Figure 3(b) shows a similar trend to the "Total Time Spent" box plots in Figure 3(a). A Friedman's test likewise shows a statistically significant effect of conditions on "Last Click Time" ($\chi^2(4) = 11.269, p = 0.0237$). Post-hoc analysis shows that the None condition ($p = 0.049$) and the S1 condition ($p = 0.038$) are significantly faster than S4. Unlike the post-hoc analysis for "Total Time Spent," we do not see statistically significant differences between None–S3 ($p = 0.766$) and S2–S4 ($p = 0.119$).



(a) Total Time Spent (b) Last Click Time

Figure 3: Box plots showing (a) average total time spent on each graph, and (b) average time of the last node click on each graph. Horizontal red and black lines denote mean and median, respectively.

6.3 Correctness Results at Identifying PENs

To measure participant correctness, we look at each person's rate of correctly identifying PENs using Jaccard index:

$$Correctness(S,G) = \frac{|S \cap G|}{|S \cup G|} \times 100\% \quad (1)$$

S denotes the set of user-selected nodes (i.e., the submitted answer) and G denotes the ground truth of PENs.

During the study, we asked the participants to make a guess if they could not identify PENs at the three minute time limit; an event that can potentially introduce false-positives to the results. We therefore remove selections made by participants after the three-minute limit.

The "Correctness (% and count)" columns in Table 3 present the detailed information about the participants' correctness rates under different conditions. The four column headers represent: (μ) the averaged correctness rates, (All) the number of participants who have correctness rates of 100%, (Some) the number of participants who have correctness rates between 1% and 99%, and (Zero) the number of participants who have correctness rates of 0%.

This distribution of correctness rates across the four columns violates the normality assumption. In fact, the distributions for each condition are U-shaped, as most participants selected either 100% of PENs (All) correctly or 0% of them correctly (None). In this scenario, standard ANOVA analysis is unsuitable. We instead perform a Friedman's test, which indicates no statistically significant effect of perceptual masking condition on correctness ($\chi^2(4) = 7.131, p = 0.1291$).

6.4 Evaluating H1–H3

Based on the statistical analyses, H1 can be accepted because both time and correctness do not present a significant difference between None and S1. For H2, there are supporting numbers for some parts of the hypothesis. That is, in terms of task completion time (both "Total Time Spent" and "Last Click Time"), None and S1 are faster than S4. However, there is no significant difference found regarding the correctness in identifying privacy. Therefore, H2 is only partially confirmed. Most parts of H3 are rejected as the statistical results only indicate a significant difference in the comparison between S2 and S4 with respect to "Total Time Spent". All other comparisons between the two groups ($\{S4\}$ and $\{S2, S3\}$) show no effect.

7 Discussion & Conclusion

At heart, this paper conducts perceptual-based testing on graph clutter. We introduce a series of techniques that perform localized

Condition	Total Time Spent (s)				Last Click Time (s)				Correctness (% and count)			
	μ	σ	Median	Rank Sum	μ	σ	Median	Rank Sum	μ	All	Some	Zero
None	122.02	48.49	113.45	61	120.86	48.33	112.93	65	70%	17	2	7
S1	128.40	50.45	134.96	65	125.58	50.63	134.96	66	65%	17	0	9
S2	142.90	46.88	168.55	79.5	136.15	46.58	151.47	77	65%	17	0	9
S3	154.68	50.45	180	89.5	149.48	38.65	171.64	87	45%	11	1	14
S4	166.26	27.45	180	95	162.27	28.92	180	96	48%	11	2	13

μ = average, σ = standard deviation

Table 3: Timing and correctness results from the user study, arranged by perceptual masking condition. Column sections show (left) the total spent needed to identify PENs, (middle) the last click time on the graph, and (right) the correctness percent and counts of clicked PENs.

violation of graph aesthetics and test how effective these violations are at visually hiding a specific set of nodes in a graph. Our use case scenario—privacy preservation—helps contextualize and motivate the study. Future work can certainly extend and expand our understanding of how perceptual masking, especially when considering more complex and scaled datasets, use of different layout techniques, the availability of user interactions, etc. This paper provides an initial reference point for such work.

7.1 Evaluating Efficacy of Perceptual Masking

To assess the efficacy of perceptual masking strategies, we analyze the performance results and ensuing analysis described in Section 6. Succinctly, perceptual masking has a stronger effect on task completion time than on correctness. As the simplest strategy, only increasing edge crossings (S1 condition) does not largely affect performance (both time and correctness) as compared to the default graph layout (None condition). However, when multiple (≥ 2) strategies are used, more time is needed to correctly identify the leaking nodes. Correctness also shows a declining trend, though not at a level that is considered statistically significant. When node clustering is involved (S3 and S4), we see the longest overall times and lowest correctness.

7.2 Study Limitations

To ensure a manageable scope and minimize potential confounding factors, we controlled several variables and introduced limitations.

As mentioned in Section 4, we do not include person-person edges in the datasets. This makes the graphs easier to standardize in terms of structure and complexity (reducing the chance for it becoming a confound), but it also means that our synthetic graphs are probably simpler than real-world ones used by domain researchers. We note however this is not always true–in some cases sociologists only look at the links between attribute nodes or represent multidimensional tabular data via ontology graphs. These types of graphs are useful for observing relationships and correlations within populations based on attributes and/or demographics. Increasing the number of edges (person-person edges or not) can only add more visual clutter, which should make privacy leaks harder to identify even without the deliberate application of masking operations.

Moreover, including person-person edges opens up new ways for identifying privacy (e.g., via techniques based on graph topology), as mentioned in Section 2. Although we do not consider cases for topology-based privacy models, the effects of perceptual masking are independent from how privacy leaks are defined.

We additionally controlled the study visualizations and interface in several other ways: similar node and edge counts, always three types of attribute nodes, the same aesthetic criteria for creating default layouts, limited and consistent user interactions, and only one *2-diversity* privacy leak for each dataset. Graph stylings were designed to be as legible as possible, including the use of semi-opaque nodes and edges to allow for easier perception of node-edge overlap and edge traversals. Varying these properties will affect users performance at identifying privacy leaks.

7.3 Subject Feedback and Future Considerations

After the study, we collected qualitative feedback from study subjects. Primarily, we were interested in their thoughts as to the efficacy of perceptual masking and what suggestions they had. We use their feedback to define a set of future considerations for designing studies on perceptual masking. Note that our study population was primarily visualization users and they did not know their study performance when asked for feedback.

Several participants commented that the graph layouts in the study were nicely structured such that nodes having the same ontology types are clearly clustered together and well-separated from the other nodes (i.e., node clustering and separation from Table 1 was followed). This does not necessarily make it simpler to identify PENs, but it does allow the participants to determine non-privacy-leaking nodes faster. One subject stated this succinctly: "*The graph layout is very structured which makes following nodes of the same group easy.*" Another participant had a similar comment: "*Because the clusters are so separate, the edges that are linked to the same group of nodes have very similar angles and directions. Edge crossings then matters less in this case, since I can still tell where they go.*"

Many participants also thought that introducing "node clusters" was the most effective strategy for masking leaks: "*I think the node clustering method is the most effective. When nodes of different groups are placed in one cluster, it increases the complexity that I need to process in my head.*" Another repeated suggestion was to render the nodes and edges with non-transparent colors: "*I think making edges and nodes opaque would be effective only when many nodes are clustered together, because that's when edge crossing and overlapping become worse. Then you add up much more complexity.*" One subject recommended to also change the rendering order: "*If you put the edges under the nodes and make them not transparent, that [task] would be much harder. Now it would look like they're connected when they're really not.*"

Synthesizing participant comments reveals possible directions and variations for evaluation of the efficacy of perceptual masking. In particular, we hypothesize that augmenting graphs in the following ways will increase the effectiveness of masking:

Make the graph layout less structured. In our study, the layout of the graphs is done semi-manually (following [5]) such that person nodes that link to the same set of ontology nodes are placed in a cluster and are distinctly separated from nodes in other clusters.

Mingle privacy leaking nodes with a larger number of non-leaking nodes. In our study, we controlled the mingled node clusters to contain less than 10 total nodes. This count can be varied, and the mingling of privacy-leaking nodes can be distributed among multiple clusters.

Update the visual encoding of the nodes and edges to be less-friendly to privacy detection. In our study, nodes and edges are rendered with semi-transparent colors. When publishing visualizations with non-transparent nodes or edges, it will become harder to trace the nodes and edges that are overlapping with each other.

7.4 Real-World Applications for Perceptual Masking

Our user study was conducted with graphs sized under 100 nodes and 260 edges. In real-world datasets, such as those collected in sociology and anthropology surveys, graphs scale much larger. Increasing graph size inherently introduces more visual clutter. This potentially makes privacy masking a more effective scheme, since the graph will have high complexity and contain more nodes and edges to visually search.

In sociology, a common strategy for hiding privacy in presented social networks is to use a hairball, despite the fact that this largely reduces the usability [5]. The perceptual masking strategies discussed in this paper allows a researcher to manipulate only regions of the graph that contain privacy leaks, thus maintaining a better overall readability of the graph.

Perceptual masking strategies introduced in this paper can be used when the visualization is shown only for a limited time—such as during presentations—where viewers cannot snapshot the chart for later reference. For occasions when a visualization image is publicly released, no time constraint can be enforced. While this may seem like a significant drawback for our study, we instead believe it should motivate future research. Perhaps there are limits where the intersection of graph complexity, graph layout, perceptual masking, etc., overcome the lack of time constraint.

7.5 Conclusion

We investigate how violating aesthetic metrics enables perceptual privacy protection for nodes in ontology network visualizations. To do this, we apply combinations of three strategies to mask privacy, each of which clutters a localized portion of the graph while maintaining the layout for the remainder of the visualization.

To evaluate the effectiveness of these techniques, we conduct a user study that tasks subjects to identify privacy leaks in a curated set of graphs. Results suggest that when multiple privacy masking operations are applied, participants take a significantly longer amount of time to identify the leaks. In particular, introducing unnecessary node clustering provides better protection than increasing edge crossings and increasing node-edge overlapping. Despite this, leaks in our study graphs can usually be eventually identified through exhaustive serial scanning.

Our work provides a reference point for future studies to evaluate privacy masking in more complex scenarios. We hypothesize these scenarios will lead to more successful perceptual masking.

Acknowledgments

This research is supported in part by the U.S. National Science Foundation through grant IIS-1320229 and IIS-1741536.

References

[1] Metrics for graph drawing aesthetics. *Journal of Visual Languages & Computing*, 13(5):501 – 516, 2002.

[2] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.

[3] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch. The aesthetics of graph visualization. In *Proceedings of the Third Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics'07, pp. 57–64, 2007.

[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, C. Bryan, and K. L. Ma. Privacy preserving visualization for social network data with ontology information. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 11–20, 2017.

[6] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.

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

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

[9] 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.

[10] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[11] M. Ghoniem, J.-D. Fekete, and P. Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Proceedings of the IEEE Symposium on Information Visualization*, pp. 17–24, 2004.

[12] C. Healey and J. Enns. Attention and visual memory in visualization and computer graphics. *IEEE transactions on visualization and computer graphics*, 18(7):1170–1188, 2012.

[13] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM International Conference on Management of Data (SIGMOD)*, pp. 93–106, 2008.

[14] 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*, pp. 954–965, 2009.

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

[16] D. McVicar and M. Anyadike-Danes. Predicting successful and unsuccessful transitions from school to work by using sequence methods. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 165(2).

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

[18] M. Pohl, M. Schmitt, and S. Diehl. Comparing the readability of graph layouts using eyetracking and task-oriented analysis. In *Computational Aesthetics*, pp. 49–56, 2009.

[19] H. G. Schutz. An evaluation of formats for graphic trend displays—experiment ii1. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 3(2):99–107, 1961.

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

[21] 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.

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

[23] M. Taylor and P. Rodgers. Applying graphical design techniques to graph visualisation. In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pp. 651–656. IEEE, 2005.

[24] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, June 2002.

[25] K. Wong and D. Sun. On evaluating the layout of uml diagrams for program comprehension. *Software Quality Journal*, 14(3):233–259, Sep 2006.

[26] L. Zhang and W. Zhang. Edge anonymity in social network graphs. In *2009 International Conference on Computational Science and Engineering*, vol. 4, pp. 1–8, 2009.

[27] 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, pp. 506–515, 2008.

[28] 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.