# Anonymising social graphs in the presence of active attackers

**Sjouke Mauw**[1,2], **Yunior Ramírez-Cruz**[2], **Rolando Trujillo-Rasua**[2,3]

[1]CSC, [2]SnT, University of Luxembourg, 6 av. de la Fonte, L-4364 Esch-sur-Alzette, Luxembourg.

[3] School of Information Technology, Deakin University, 221 Burwood Hwy., Burwood VIC 3125, Australia.

E-mails: {sjouke.mauw, yunior.ramirez}@uni.lu, rolando.trujillo@deakin.edu.au

**Abstract.** The publication of social network graphs enables researchers to understand and investigate human behaviour. However, when such analyses can target single individuals rather than society as a whole, it is clear that privacy becomes a serious concern. This article addresses the challenge of publishing social graphs with proven privacy guarantees. In our adversarial model we consider an *active attacker*, who has the capability of altering the structure of the social graph before publication by creating new user profiles and establishing relations between them and the targeted network users. We aim to protect graphs satisfying $(1,1)$-anonymity, the privacy property that characterises the weakest graphs in terms of resistance to active attacks. To that end, we introduce a class of perturbation methods based on edge additions that transform a $(1,1)$-anonymous graph into a graph satisfying $(k,\ell)$-anonymity for some $k > 1$ or some $\ell > 1$. We prove the correctness of our approach and give a tight upper bound on the number of necessary edge additions. Experimental results, obtained on real-life graphs and a large collection of randomly generated graphs, show that our methods effectively prevent attacks from active adversaries with the capability of adding one node to the network, and additionally provide some level of protection against more capable attackers. We also conducted experiments on real-life social graphs which show that the outputs of state-of-the-art community detection algorithms on the anonymised graphs is similar to those obtained on the original graphs.

**Keywords.** privacy, social networks, active attacks, $(k,\ell)$-anonymity

## 1 Introduction

In recent decades, human interaction and socialisation has dramatically changed, propelled by new developments on communication and information technologies. With the advent of online social networks (OSN), emotions, feelings, thoughts, can all be shared instantly. The feeling that online interactions are to some extent less consequential than face-to-face communication eases the process of deciding what to share, how to share it, and to what audience. Thus, OSNs make it easier to disclose personal information, as users typically feel more protected from the public when interacting through a computer.

The success of OSNs has brought about the availability of an enormous amount of information, among which social graphs are especially useful. Informally, a social graph is a static representation of a social network at a given moment. Every vertex corresponds to a user and edges represent relations such as friendship, shared work experience, common affiliations, etc. Using graph theory and methods from modern sociology, researchers are able to obtain useful knowledge regarding the properties of the network and the behaviour of the users. Typical social network analysis tasks are, among others, community detection, which allows to detect groups of users that display a common behaviour or are somehow strongly interrelated; link prediction, commonly used for suggesting new friends; and centrality analysis, which helps to determine the role a user plays in the network.

The owners of OSN systems occasionally release samples of their social graphs, whether as a tool to help advance research, as a self-advertising mechanism, etc. In general, the possibility to discover new knowledge is positively viewed by the public, but it is not acceptable that the privacy of social network users is compromised in the process. For example, a malicious agent or entity, usually referred to as *adversary*, may attempt to identify a user in a published social graph, an action often called *re-identification*, and learn sensitive information such as political and religious preferences. To prevent this risk, social graphs must be properly anonymised before releasing them.

A straightforward anonymisation technique that may immediately come to mind consists in removing identifying attributes from the graph, such as name, email address, and social security number. However, even a simple graph without attributes attached to its vertices can be subject to re-identification attacks [14]. For example, an adversary who knows the number of social links of a target victim can use this information to re-identify the victim, if there is a unique node with this number of links.

Re-identification attacks to social graphs are typically categorised as *passive* or *active*. In a passive attack, the adversary's actions aiming to re-identify the victims are performed only after the social graph has been released, and usually by means of information collected from publicly available sources, e.g., other OSNs sharing a part of the user base [14]. On the other hand, in an active attack the adversary proactively inserts new nodes in the network, which are referred to as *sybil nodes*, and tries to establish links with the targeted victims before the graph is released. The links are made in such a way that every victim connects to the set of sybil nodes in a unique and re-identifiable manner. Once the social graph is released, the adversary first identifies her own set of sybil nodes, which are subsequently used to re-identify the victims by using their unique connection patterns to the set of sybil nodes [1, 23].

Active attackers are stronger than passive ones, in the sense that they can create the structural patterns needed for re-identification, instead of expecting the patterns to naturally occur. However, this type of attacks has not received sufficient attention so far. The first privacy property that accounts for active attackers was recently proposed in [20]. This property, which is called $(k,\ell)$-anonymity, characterises social graphs where a user cannot be re-identified with probability higher than $1/k$ by an active attacker who is able to enrol up to $\ell$ sybil nodes. It was shown in [20] that real-life social graphs tend to be $(1,1)$-anonymous, which is the lowest privacy level possible, as it means that there exists at least one vertex in the graph uniquely identifiable by its distance to some other vertex. This leads to the question addressed in this paper, namely whether it is possible to define privacy-preserving transformation techniques that counteract active attacks by transforming a graph with low anonymity, in terms of $(k,\ell)$-anonymity, into a graph with higher anonymity guarantees against re-identification by active attackers.

**Contributions.** This article introduces a class of perturbation methods to transform a $(1,1)$-anonymous graph $G$ (the least private graph in terms of $(k,\ell)$-anonymity) into a graph $G'$ that satisfies $(k,\ell)$-anonymity for some $k>1$ or some $\ell>1$. All methods in the class are based on performing edge-additions until the desired anonymity property is achieved. Additionally, we formally prove that all methods in this family have a common theoretical upper bound on the number of added edges. Providing such a family of methods, all of which offer the same privacy guarantee, allows the data holder to choose the one that least perturbs the original data, for some interpretation of perturbation. The results presented in this article generalise and extend our previous work reported in [11], as the proposed class contains the anonymisation method introduced in that paper.

We provide comprehensive empirical validation of our results on both real-life and randomly generated graphs. Three real-life graphs, as well as a large collection of random graphs, are used to show the resistance of the anonymised graphs to a paradigmatic active attack: the *walk-based attack* described in [1]. Then, real-life social graphs are used to show that, while the proposed methods provide equivalent resistance to active attacks, they slightly differ in terms of perturbation. We define perturbation as the difference between the original and the anonymised graph with respect to some utility measure. The results of these experiments lead to two main conclusions. Firstly, our approach introduces small perturbations in terms of the number of added edges, some global graph parameters, and the variation of the outputs of state-of-the-art community detection algorithms. Secondly, no individual method from our class universally outperforms the remaining class members in terms of all considered utility measures. Thus, in practical settings, it is the task of the practitioner to choose the appropriate instance according to the input graph and the considered utility criterion.

**Structure of the paper.** The remainder of this paper is structured as follows. Section 2 provides the background on privacy-preserving publication of social graphs, explaining in detail privacy properties geared towards passive and active adversaries. A thorough discussion on $(k, \ell)$-anonymity, and its role in protecting social graphs from active attacks is provided in Section 3. Then, Section 4 presents and proves properties of $(1, 1)$-anonymous graphs, which form the theoretical foundation of our anonymisation strategy (also described in this section). The correctness and convergence of our methods, as well as computational complexity issues, are treated in Section 5, whereas Section 6 presents the empirical evaluation of the proposed methods on randomly generated and real-life social graphs. Finally, conclusions are drawn in Section 7.

## 2 Related work

Since 2002, a large body of work in data anonymisation has spun around the notion of $k$-anonymity [17]. This idea, which appeared in the field of microdata anonymisation, has been ported to the field of social graphs. Here, we will briefly cover this notion in its original background, and then analyse the manner in which it has been ported to our domain of interest. While covering the application of $k$-anonymity notions to social graphs, we will distinguish those approaches focused on passive attacks from the ones relating to active attacks.

**$k$-anonymity in microdata.** A pioneer study on re-identification attacks was published in 2002 by Sweeney [19]. Sweeney estimated that 87% of the population in the United States can be uniquely identified by combining publicly available attributes such as gender, date of birth and zip code, even if every such attribute is not unique in the population if taken independently.

The potential success of an adversary strongly depends on the amount and nature of the background knowledge used to uniquely characterise the victims. Such knowledge may come from public sources (e.g. census data), data brokers, or may be collected through malicious actions. This background knowledge is used to search for unique matches in released databases, which allows for re-identification of the victims. Thus, the publisher of the data must be careful to assure that no entry can be uniquely identified, regardless of the background knowledge used by the adversary. That was the setting in which *$k$-anonymity* was proposed as a measure to quantify the adversary's re-identification capability in the worst case scenario [17].

A dataset is said to satisfy $k$-anonymity if every record is indistinguishable from $k-1$ other records with respect to a given adversary's background knowledge. That is, $k$-anonymity ensures that the probability of the adversary re-identifying the user represented by an individual record is at most $1/k$. Given that a large number of legitimate data analysis tasks seek to describe general trends on large populations instead of characteristics of particular individuals, datasets can be processed in order to attain $k$-anonymity while still remaining useful for analysis.

Consider, for example, a table where each record contains date of birth, zip code, and some health conditions suffered by a person. The combination of date of birth and zip code may be unique for some users, thus allowing the information about their health conditions to be disclosed, with all the negative implications this would entail. However, it is possible to modify the table by replacing date of birth by a more general information, e.g. year of birth, or month-year, and replacing the zip code by a higher level geographical entity such as city or county. The modified table would still contain completely truthful information and, at the same time, is likely to satisfy $k$-anonymity for some $k > 1$, while additionally being useful for numerous analysis tasks, e.g. studying the evolution of a disease at state level during several years.

**$k$-anonymity in social graphs.** Social network users provide a wealth of personal information that can be stored and processed as traditional microdata, and as such it is susceptible to the same types of re-identification attacks and can be processed by similar anonymisation techniques. However, the information on relations contained in social graphs allows the adversaries to enrich their background knowledge in such a way that even simple graphs which have been stripped of all user information can still be used to uniquely identify users, thereby disclosing potentially sensitive information about their relationships, political or religious affiliations, etc. In 2009, Narayanan et al. collected simple, publicly available information about relationships between Flickr users and were able to conduct a passive attack and to re-identify one third of the users on a naively anonymised Twitter graph with only a 12% error rate [14]. It is worth noting that not all Twitter users involved in the experiment had Flickr accounts, and *vice versa*.

In order to tackle this problem, several graph-oriented notions of $k$-anonymity have been proposed, considering the nature of the knowledge used by passive attackers. In social graphs, adversary knowledge is normally modelled using structural properties on the graph, e.g. the degree of a vertex, the topology of the subgraph induced by the neighbourhood of a vertex, etc. Under the specified properties, two vertices are said to be indistinguishable if they are equivalent with respect to the considered structural property. For example, Liu et al. [10] introduced the notion of $k$-degree anonymity. A graph is said to be $k$-degree anonymous if for every vertex of the graph there exist $k-1$ other vertices with the same degree. In the same paper, Liu et al. devised a simple and efficient algorithm to transform a graph into a $k$-degree anonymous graph. The authors also show that the utility of the graph is at least partly kept, because the average path length and the exponent of the power-law distribution of the original graph are preserved.

Examples of other structural privacy properties are $k$-neighbourhood anonymity [29] and $k$-automorphism anonymity [6, 30]. In the first case, for every vertex $v$ in the graph there must exist at least $k-1$ other vertices $v_1, \ldots, v_{k-1}$ such that the subgraph induced by $v$'s neighbours is isomorphic to the subgraph induced by $v_i$'s neighbours, for every $i \in \{1, \ldots, k-1\}$. In the second case, two vertices $u$ and $v$ are considered to be equivalent if there is an automorphism of the graph where $u$ maps to $v$. As the authors discuss, it is very unlikely for real-life social graphs to satisfy $k$-automorphism anonymity, whereas graphs perturbed in order to satisfy the property tend to be useless for analysis [30].

**Active attacks.** The privacy notions described above assume that the adversary has the capability of gathering information from public sources, but do not consider the scenario where the adversary is also able to create user accounts and interact with the OSN users in order to induce the creation of structural patterns that facilitates their re-identification.

Backstrom et al. were the first to show the feasibility of active attacks in social networks [1]. They proposed several attacks where the adversary inserts a number of nodes in the social graph and establishes a connection pattern among them that ensures that, with high probability, the induced subgraph can be uniquely identified and efficiently retrieved once the graph has been released. The relations between the nodes in the adversary's subgraph and the victim nodes are established in such a way that every victim ends up having a unique *fingerprint* of links to the adversary's subgraph. Once the social graph is published, the adversary retrieves the planted subgraph and re-identifies those nodes that preserve the expected fingerprints.

The main challenge the adversary has to face when conducting an active attack is the existence of sybil detection techniques [26], which identify and counteract the anomalous behaviours of the sybil nodes. Thus, the adversary has to pursue the ability to compromise as many victims as possible while keeping the set of sybil nodes unnoticed. In order to reduce the number of sybil nodes needed, some hybrid attacks have been proposed. Hybrid attacks rely on small sets of sybil nodes, called *seeds*, and a combination of active and passive techniques. Wei et al. [23] introduced the so-called *Seed-and-Grow* attack, which relies on a small set of sybil nodes to re-identify an initial set of victims, and then uses the information from an auxiliary, non-anonymised graph, to re-identify some of the victims' neighbours. This process is iteratively repeated, using at every step the set of vertices that were re-identified in the previous step. Note that the success of the passive phase of the attack depends on the success of the active part, so if the latter is effectively countered the entire attack fails.

Privacy-preserving publication of social graphs in the presence of active attacks is a challenging task. In fact, none of the privacy properties described above [10, 29, 6, 30] adequately captures the ability to effectively counteract active attacks. As pointed out in [7], the defence strategy against sybil nodes has focused on the detection and removal of such nodes, whereas the ability to successfully anonymise the potentially attacked graphs has been overlooked. In fact, the methods referred to in the literature as *near-optimal sybil defences* [7, 27, 28] do not actually attempt to remove all sybil nodes, but to lower-bound their number to around $O(\log n)$, where $n$ is the number of nodes in the network. However, as pointed out by Backstrom et al., for some active attacks (e.g. the walk-based attack that we treat later on) this number of sybil nodes is sufficient for being able to re-identify most legitimate users, with high probability [1]. Thus, a graph that has been compromised by such an attack may still be considered as safe by sybil defences and have no action taken on them. Finally, we must take into account that an active attack can be mounted by initially honest users who turn rogue and collude to compromise other users in the network. This type of attack can hardly be prevented by sybil detection techniques, as the malicious behaviour has been limited in time.

To the best of our knowledge, the first privacy measure to evaluate the resistance of social graphs to active attacks is $(k, \ell)$-anonymity, proposed in [20]. Trujillo-Rasua and Yero model the adversary's background knowledge about each vertex as a vector, called the *metric representation*, which contains the distances from that vertex to each sybil node planted by the adversary. This allows $(k, \ell)$-anonymity to measure users' privacy against an active adversary capable of retrieving its own attacker subgraph. The notion of $(k, \ell)$-anonymity does not consider the use of sybil detection. Instead, it assumes that any vertex subset of cardinality up to $\ell$ can potentially be a set of sybil nodes and models the protection of every other vertex from them.

In $(k, \ell)$-anonymity, the parameter $k$ represents a privacy threshold as in the standard $k$-anonymity notion. That is, the considered adversary is unable to re-identify a targeted victim with probability higher than $1/k$ in a graph satisfying $(k, \ell)$-anonymity. The parameter $\ell$ is an upper bound on the number of sybil nodes that the adversary is judged capable of introducing into the network. Expressed differently, $\ell$ is a bound on the estimated adversary capability to insert sybil nodes without being detected.

As we discussed previously, this paper focuses on several graph transformations aimed at improving privacy in terms of $(k, \ell)$-anonymity. To facilitate readability, we provide in the next section a formal definition for this privacy notion and introduce the most important terminology and notation that we use throughout the article.

## 3 Preliminaries

We model a social graph $G = (V, E)$ as a simple graph, i.e. a graph containing no loops or multiple edges, where the vertex set $V$ represents the set of users and the edge set $E$ their relationships. The notion of $(k, \ell)$-anonymity [20] assumes that the adversary can use the distances between the sybil nodes and the victims for re-identification. The distance between two vertices $u$ and $v$ of $G$, denoted by $d_G(u, v)$ (or simply as $d(u, v)$ if there is no ambiguity), is understood as the number of edges in a shortest path connecting $u$ and $v$.

As discussed in [20], active attackers seek to introduce a set of sybil nodes in the social network and make every victim identifiable by means of a unique structural fingerprint, which is constructed using its distances from each sybil node. In order to encode this knowledge, they propose to use the so-called *metric representation* [4, 18] as the structural property that represents the adversary's background knowledge in active attacks.

**Definition 1** (Metric representation [4, 18])**.** Let $G = (V, E)$ be a simple connected graph and let $R = (u_1, ..., u_t)$ be a vector whose components are vertices of $G$. The *metric representation* of a vertex $v$ with respect to $R$ is the vector

$$r(v|R) = (d_G(v, u_1), \dots, d_G(v, u_t)).$$

From the attacker's perspective, every vertex having a unique metric representation with respect to the sybil nodes (in some particular order) can be re-identified after the graph is published and the set of sybil nodes is retrieved. That imposes on the data holder the

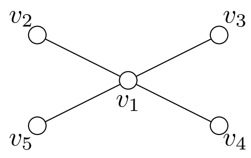

Figure 1: A star graph.

necessity of making sure that, when the graph is published, every vertex is rendered indistinguishable from a sufficiently large number of other vertices. That is, given a set of sybil nodes $S$ and the vector $R_S$ obtained by imposing some ordering on $S$, the data holder must ensure that the metric representation of every vertex with respect to $R_S$ is not unique. This property is captured in [20] by the notion of *k-antiresolving*[1] *set*, which is enunciated as follows.

**Definition 2** ($k$-antiresolving set [20])**.** Let $G = (V, E)$ be a simple connected graph and let $S = \{u_1, \ldots, u_t\}$ be a subset of $V$. Let $R_S = (u_{i_1}, \ldots, u_{i_t})$ be a vector composed of the elements from $S$ in some order. The set $S$ is called a *k-antiresolving set* of $G$ if $k$ is the greatest positive integer such that for every vertex $v \in V \setminus S$ there exist at least $k-1$ vertices $v_1, \ldots, v_{k-1} \in V \setminus S$ such that $v, v_1, \ldots, v_{k-1}$ are pairwise different and satisfy

$$r(v|R_S) = r(v_1|R_S) = \ldots = r(v_{k-1}|R_S).$$

Note that while Definition 2 requires that some ordering can be imposed on $V$, the nature of this ordering itself is not relevant: it can be arbitrary, as long as it is consistently used. Thus, from now on we will simply assume that such an order exists (it always does) and that it is always the one imposed.

It is simple to see that if the data holder manages to ensure that the set of sybil nodes is $k$-antiresolving, then every possible victim has the same metric representation as at least $k-1$ other vertices, so the re-identification probability is at most $1/k$. As an example, consider the star graph in Figure 1. The distance from $v_1$ to any other vertex in the graph is 1, thus $\{v_1\}$ is a 4-antiresolving set. On the other hand, any set $\{v_i\}$ with $i \in \{2,3,4,5\}$ is a 1-antiresolving set because $r(v_1|(v_i)) = (1)$ while $r(v_j|(v_i)) = (2)$ for every $j \in \{2,3,4,5\}$ and $j \neq i$. Finally, we consider the subset $\{v_1, v_5\}$. We observe that $r(v_2|(v_1,v_5)) = r(v_3|(v_1,v_5)) = r(v_4|(v_1,v_5)) = (1,2)$, implying that $\{v_1, v_5\}$ is a 3-antiresolving set.

The notion of $k$-antiresolving set is sufficient for modelling the privacy of every vertex with respect to one given set of sybil nodes, but it is still insufficient for protecting the graph from an active attacker. Since the attacker will do her best to ensure that the set of sybil nodes goes unnoticed to sybil detection mechanisms, one cannot assume that the set of sybil nodes will be known. Acknowledging this problem, the notion of $(k,\ell)$-anonymity takes into consideration the possibility that any subset of vertices of size up to $\ell$ is the set of sybil nodes introduced by the attacker. Thus, a graph $G$ satisfying $(k,\ell)$-anonymity ensures that every subset of vertices with cardinality at most $\ell$ is a $k'$-antiresolving set for

[1]The term *antiresolving* comes in opposition to the notion of *resolving set* introduced in 1976 by Harary and Melter [4] (it had been independently introduced in 1975 by Slater [18] as *locating set*). Given a graph $G = (V, E)$, a set $S \subseteq V$ is a resolving set of $G$ if for every pair of different vertices $x, y \in V \setminus S$ there exists $z \in S$ such that $d(x,z) \neq d(y,z)$. From this definition it easily follows that a resolving set is a 1-antiresolving set. The implication does not hold in the other direction, though.

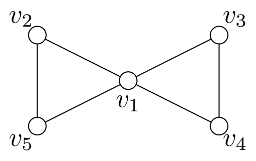

Figure 2: An anonymised version of the star graph in Figure 1. This graph satisfies $(2, 1)$-anonymity.

some $k' \geq k$. In other words, every vertex in $G$ is indistinguishable from at least $k - 1$ other vertices in terms of their metric representations with respect to any subset of vertices of cardinality at most $\ell$. That means that a $(k, \ell)$-anonymous graph bounds the re-identification capacity of an attacker leveraging up to $\ell$ sybil nodes to at most $1/k$ for every possible victim, regardless of the identity of the sybil nodes inserted in the graph.

In order to give the formal definition of $(k, \ell)$-anonymity, it is first necessary to introduce the notion of $k$-metric antidimension [20], which characterises a graph in terms of the sizes of its smallest $k$-antiresolving sets.

**Definition 3** ($k$-metric antidimension [20] )**.** The *$k$-metric antidimension* of a simple connected graph $G = (V, E)$ is the minimum cardinality amongst the $k$-antiresolving sets of $G$.

Considering again the star graph depicted in Figure 1, we observe that the singleton set $\{v_2\}$ is a 1-antiresolving set of this graph. Ergo, the 1-metric antidimension of this graph is 1. In order to determine the 2-metric antidimension, first notice that $v_1$ should be included in any 2-antiresolving set, while $\{v_1\}$ itself is a 4-antiresolving set. Therefore, the 2-metric antidimension of the star graph is greater than or equal to 2. However, the subset $\{v_1, v_i\}$ for every $i \in \{2, 3, 4, 5\}$ is a 3-antiresolving rather than a 2-antiresolving set. Consequently, the 2-metric antidimension of the graph in Figure 1 is 3, given that $\{v_5, v_1, v_3\}$ is a 2-antiresolving set. Finally, we enunciate the formal definition of $(k, \ell)$-anonymity, as provided in [20].

**Definition 4** ($(k, \ell)$-anonymity [20])**.** A graph $G$ is said to satisfy $(k, \ell)$-anonymity if $k$ is the smallest positive integer such that the $k$-metric antidimension of $G$ is lower than or equal to $\ell$.

Re-taking our running example, note that the graph shown in Figure 1 is $(1, 1)$-anonymous, as the singletons $\{v_2\}$, $\{v_3\}$, $\{v_4\}$ and $\{v_5\}$ are 1-antiresolving sets.

Summing up the previous discussion, the purpose of the data holder when publishing a social graph in the presence of active attackers is to release a $(k, \ell)$-anonymous graph, where $\ell$ is an upper bound on the number of sybil nodes that an adversary is judged capable of inserting in the graph, and $k$ is the minimum size of the vertex subsets within which the identity of each particular vertex is hidden. In the case where the original graph is $(1, 1)$-anonymous, as in our running example, the need of an anonymization method becomes apparent. For example, instead of publishing the original graph in Figure 1, the data holder may release a similar graph that satisfies $(k, \ell)$-anonymity for some $k > 1$ or some $\ell > 1$, such as the graph in Figure 2.

## 4 Protecting $(1,1)$-anonymous graphs

According to Definition 4, the weakest privacy property for a graph facing active attacks is $(1,1)$-anonymity. Indeed, in a $(1,1)$-anonymous graph, there is at least one 1-antiresolving set of cardinality 1, which means that at least one user can be successfully re-identified by an active adversary leveraging a single sybil node. For this reason, in this paper we set up the goal of protecting $(1,1)$-anonymous graphs. To that end, we have devised a family of methods that transform a $(1,1)$-anonymous graph $G$ into a graph $G'$ that satisfies $(k,\ell)$-anonymity for some $k > 1$ or some $\ell > 1$. As we will show later, the transformations performed by our methods effectively protect any graph against active attackers with the ability to insert one sybil node in the network, while additionally increasing to some extent the level of protection against stronger active attackers.

Graph anonymity is typically achieved through successive graph transformation operations, such as adding and/or removing vertices/edges, until the resulting graph satisfies the desired privacy property. While these different types of operations can be used in combination, most anonymisation techniques focus on a single type of operation. For example, the pioneer work by Liu and Terzi [10] tackles the problem of achieving $k$-degree anonymity by relying on edge addition operations only. Chester et al. [2], instead, address the same problem by adding vertices to the original graph. In this article we focus on edge addition operations, and the resulting anonymisation problem is defined as follows.

**Problem statement** (Graph anonymisation via edge-addition)**.** Given a graph $G = (V, E)$, find a subset $S \subseteq (V \times V) \setminus E$ of minimum cardinality such that $G' = (V, E \cup S)$ is $(k,\ell)$-anonymous with $k > 1$ or $\ell > 1$.

To the best of our knowledge, there does not exist an efficient algorithm to solve the problem above. Indeed, as most anonymisation problems based on $k$-anonymity [5, 13], this problem is likely to be NP-hard. Therefore, we will develop a generic algorithm to approximate the solution to this problem. The remainder of this section is thus focused on proving theoretical properties of $(1,1)$-anonymous graphs, which we use to prove the correctness and convergence of our class of anonymisation methods.

### 4.1 Properties of $(1,1)$-anonymous graphs

If a graph $G$ contains a 1-antiresolving set $\{v\}$, then there exists a vertex $u$ such that $d(v,u) \neq d(v,w)$ for every $w \in V \setminus \{v,u\}$. Following terminology from [20], we call such a vertex $u$ a 1-*resolvable* vertex, in particular, we say that $u$ is 1-*resolvable by* $\{v\}$. It follows that containing a 1-resolvable vertex is a sufficient and necessary condition for a graph $G$ to be $(1,1)$-anonymous.

**Proposition 5.** *[11][2] A simple connected graph $G = (V, E)$ satisfies $(1,1)$-anonymity if and only if it contains a 1-resolvable vertex.*

*Proof.* If $G$ contains a 1-resolvable vertex $v$, then there exists a vertex $u$ in $G$ such that $\{u\}$ is a 1-antiresolving set. In consequence, $G$ is $(1,1)$-anonymous. Now, let us assume that $G$ is $(1,1)$-anonymous and that there does not exist a 1-resolvable vertex in $G$. This implies that there does not exist a 1-antiresolving set of cardinality 1 in $G$. Therefore, if a 1-antiresolving set in $G$ exists then $G$ is $(1,\ell)$-anonymous for some $\ell > 1$, otherwise $G$ is $(k,\ell)$-anonymous for some $k > 1$. In either case $G$ is not $(1,1)$-anonymous, which is a contradiction. □

[2]We cite the conference version [11] of this article whenever a result directly taken from that paper is shown.

Since the presence of 1-resolvable vertices implies $(1,1)$-anonymity, we are interested in finding those vertices in the graph which are 1-resolvable. A first simple result in this direction is the following, which describes the case of graphs having end-vertices, that is, vertices with degree one.

**Lemma 6.** *[11] For every end-vertex $v$ in a graph $G = (V, E)$ it holds that $v$'s neighbour is 1-resolvable by $\{v\}$.*

*Proof.* We should first notice that if $|V| = 2$ then both $v$ and $v$'s neighbour are 1-resolvable. Thus, let us assume that $|V| > 2$ and let $u$ be $v$'s neighbour. Because any path to $v$ passes through $u$, we obtain that $d(w, v) = d(w, u) + d(u, v) > d(u, v) = 1$ for every $w \in V \setminus \{v, u\}$. Therefore, $\{v\}$ is a 1-antiresolving set and $u$ is a vertex 1-resolvable by $\{v\}$. $\square$

A consequence of Lemma 6 is that every graph with end-vertices is $(1,1)$-anonymous. Lemma 6 is in fact a consequence of the following result, which allows us to locate all vertices that are 1-resolvable by some 1-antiresolving singleton set. In order to introduce this result, we first give some necessary definitions. The *eccentricity* $\epsilon_G(v)$ of a vertex $v$ in a connected graph $G$ is the greatest number of edges in a shortest path between $v$ and any other vertex in $G$. We call such a shortest path an *eccentricity path of* $v$. If the vertex $v$ is an end-vertex, then $v$'s neighbour lies in every eccentricity path of $v$. We prove next that, indeed, every vertex that is 1-resolvable by $\{v\}$ lies in an eccentricity path of $v$.

**Lemma 7.** *[11] Let $G$ be a simple connected graph, let $\{v\}$ be a 1-antiresolving set in $G$, and let $v_1 \dots v_m$ be an eccentricity path of $v$, i.e., $v_1 = v$. For every vertex $u$ that is 1-resolvable by $\{v\}$ there exists $i \in \{1, \dots, m\}$ such that $u = v_i$.*

*Proof.* Let us assume that $u \neq v_i \; \forall i \in \{1, \dots, m\}$. By definition, the eccentricity of $v$ satisfies that $\epsilon(v) \geq d(v, w)$ for every $w \in V(G)$ and, in particular, $\epsilon(v) \geq d(v, u)$. Given that $d(v, v_m) = \epsilon(v) \geq d(v, u)$, there must exist $i \in \{1, \dots, m\}$ such that $d(v, u) = d(v, v_i)$ (see Figure 3). Consequently, either $u = v_i$ or $u$ is not 1-resolvable by $\{v\}$, which both lead to a contradiction. $\square$

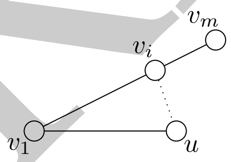

Figure 3: An eccentricity path $v_1 \dots v_i \dots v_m$ of $v_1$ and a vertex $u$ located out of that path.

## 4.2 Our anonymisation strategy

Lemma 7 provides the grounds for our anonymisation strategy, which consists of iteratively finding vertices $v$ such that $\{v\}$ is a 1-antiresolving set of $G$, determining the set of vertices that are 1-resolvable by $\{v\}$ (which all lie on an eccentricity path $v_1 \dots v_m$ of $v$), and modifying $G$ in such a way that every vertex $w \in \{v_1, \dots, v_m\}$ is not 1-resolvable by $\{v\}$ in the modified graph. As a preprocessing step, we add an edge from each end-vertex $u$ of $G$ to a randomly chosen vertex $v$ satisfying $d_G(u, v) = 2$. Thus, in what follows we will provide results on graphs which do not contain end-vertices. We set up the goal of anonymising all vertices 1-resolvable by $\{v\}$ by means of adding a single edge to the graph, which leads to the notion of *anonymising edge*.

**Definition 8** (Anonymising edge)**.** Given a graph $G = (V, E)$, an edge $(x, y) \in (V \times V) \setminus E$ is said to be an *anonymising edge* if there exists a 1-antiresolving set $\{v\}$ of $G$ such that for every vertex $u$ that is 1-resolvable by $\{v\}$ in $G$, it holds that $u$ is *not* 1-resolvable by $\{v\}$ in $G' = (V, E \cup \{(x, y)\})$.

The action of adding one anonymising edge satisfies the property that, for some 1-antiresolving set $\{v\}$, the resulting graph does not contain any vertex that was 1-resolvable by $\{v\}$ in the original graph and continues to be 1-resolvable by $\{v\}$ in the new graph. The successive obfuscation of 1-resolvable vertices is the core of the generic definition of our anonymisation strategy, which is depicted in Algorithm 1.

**Algorithm 1** Generic framework for transforming a graph $G = (V, E)$ into a graph $G' = (V, E')$ that is $(k, \ell)$-anonymous for some $k > 1$ or some $\ell > 1$.

```
1: Let $G'$ be the result of adding the necessary edges to $G$ to make it end-vertex free.
2: while $G'$ contains 1-resolvable vertices do
3:     Find a set $S$ of anonymising edges in $G'$    ▷ by, for example, using Algorithm 2 below
4:     Modify $G'$ by adding an edge from $S$    ▷ Sec. 6 gives various selection techniques
5: end while
6: return $G'$
```

Our strategy consists in adding, at each iteration, one edge to the graph until the resulting graph becomes $(k, \ell)$-anonymous for some $k > 1$ or some $\ell > 1$. Because the added edge is an anonymising edge, the graph transformation satisfies that one or more 1-resolvable vertices are obfuscated. Note that the iterative process presented in [11], where the edge addition performed at each step was called *$v$-transformation*, is an instantiation of this generic strategy.

Every instantiation of Algorithm 1 must provide procedures for making the initial graph end-vertex free and finding a set of anonymising edges, and a criterion to select an edge from this set. The first issue is trivial. As we mentioned above, we make a graph end-vertex free by adding an edge from each end-vertex to a randomly chosen vertex at distance 2 from it. We address the third issue in Section 6, where we provide different criteria that can be used to choose an edge from an anonymising edge set.

Regarding the second issue, it can be tackled by exhaustive search. That is, one can test every edge in $(V \times V) \setminus E$ to check whether it is an anonymising edge. The advantage of an exhaustive search is that it finds all anonymising edges. A major drawback, however, is its high computational complexity. Because every edge addition requires the re-computation of all shortest paths in the resulting graph, and the process of verifying whether an edge is an anonymising edge has $\mathcal{O}(|V|^2)$ time complexity, the exhaustive search approach has $\mathcal{O}(|(V \times V) \setminus E| \times |V|^3 \times |V|^2)$ computational complexity, which becomes $\mathcal{O}(|V|^7)$ in low/medium density graphs. We show next how to determine a non-trivial set of anonymising edges in $\mathcal{O}(|V|^2)$-time, assuming that all shortest paths in the original graph have already been computed.

## 4.3 Determining a set of anonymising edges

The next result introduces a property of a particular family of unicyclic graphs, on which we rely to find edges that can be added to an anonymising edge set. Such family is known as *tadpole* graphs. An $(h,t)$-*tadpole* is the result of taking a cycle graph $C_h$ of order $h$, a path graph $P_t$ of order $t$, and joining both by adding an edge between an arbitrary vertex of $C_h$ and one of the ends of $P_t$. Note that, for the remainder of this paper, we will use the notation $\delta_G(v)$ (or simply $\delta(v)$ if there is no ambiguity) for the degree of a vertex $v$ in a graph $G$.

**Theorem 9.** *Let $G$ be an $(h,t)$-tadpole. Let $u$ be the sole degree-3 vertex of $G$ and let $v$ be one of the neighbours of $u$ in the cycle subgraph of $G$. The following results hold:*

*i. If $h$ is odd, then $\{v\}$ is a 2-antiresolving set of $G$ if and only if $t \leq \frac{h-3}{2}$.*

*ii. If $h$ is even, then $\{v\}$ is a 2-antiresolving set of $G$ if and only if $t = \frac{h-2}{2}$.*

*Proof.* Let $G$ be the $(h,t)$-tadpole obtained from the cycle graph $C_h$ and the path graph $P_t$, and let $w$ be the neighbour of $v$ different from $u$. We will first assume that $h$ is odd (see Figure 4(a) for reference). Since $h$ is odd, for every vertex $x \in V(C_h) - \{v\}$ there exists $y \in V(C_h) - \{v,x\}$ such that $d(v,x) = d(v,y)$, as $v$ has two diametral vertices in $C_h$. Note also that every $x \in V(C_h)$ satisfies $d(v,x) \leq \frac{h-1}{2}$, as $h$ is odd. Finally, we have that every vertex $z \in V(P_t)$ satisfies $d(v,z) = d(v,u) + d(u,z) \leq 1 + t$. Thus, if $t \leq \frac{h-3}{2}$, the vertex $z$ satisfies $d(v,z) \leq 1 + \frac{h-3}{2} = \frac{h-1}{2}$, so there exists $x \in V(C_h) - \{u,v,w\}$ such that $d(v,z) = d(v,x)$. In consequence, we have that $\{v\}$ is a 2-antiresolving set.

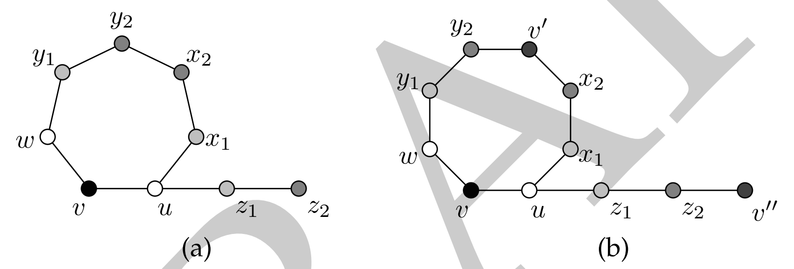

Figure 4: Two tadpole graphs. Equal grey levels indicate equidistance from the vertex $v$. (a) A $(7,2)$-tadpole. (b) A $(8,3)$-tadpole.

Let us now assume that $\{v\}$ is a 2-antiresolving set of $G$ and $t > \frac{h-3}{2}$. Consider the vertex $z' \in P_t$ such that $\delta_G(z') = 1$, that is $z'$ is the sole end-vertex of $G$, and let $x \in V(G) \setminus \{z'\}$. On the one hand, if $x \in V(P_t) \setminus \{z'\}$, clearly $d(v,x) < d(v,z')$. On the other hand, if $x \in V(C_h)$, then $d(v,x) \leq \frac{h-1}{2} < d(v,z')$. Consequently, $z'$ is 1-resolvable by $\{v\}$, which is a contradiction, so we can conclude that $\{v\}$ being a 2-antiresolving set of $G$ implies $t \leq \frac{h-3}{2}$.

We now address the case where $h$ is even (see Figure 4(b) for reference). Let $v'$ be the vertex antipodal to $v$ in $C_h$ and let $v''$ be the sole end-vertex of $G$. If $t = \frac{h-2}{2}$, we have that $d(v,v') = d(v,v'')$. Moreover, for every $x \in V(C_h) \setminus \{v,v'\}$ there exists a vertex $y \in V(C_h) \setminus \{v,v',x\}$ such that $d(v,x) = d(v,y)$; whereas for every $z \in V(P_t) \setminus \{v''\}$ there exists $x \in V(C_h) \setminus \{v,v'\}$ such that $d(v,x) = d(v,z)$. Thus, $\{v\}$ is a 2-antiresolving set of $G$.

Now let $\{v\}$ be a 2-antiresolving set of $G$ and assume that $t \neq \frac{h-2}{2}$. As before, let $v'$ be the vertex antipodal to $v$ in $C_h$ and let $v''$ be the sole end-vertex of $G$. If $t < \frac{h-2}{2}$, then $v'$ is 1-resolvable by $\{v\}$, whereas if $t > \frac{h-2}{2}$, then $v''$ is 1-resolvable by $\{v\}$. In both cases we reach a contradiction, so we can conclude that $\{v\}$ being a 2-antiresolving set of $G$ implies $t = \frac{h-2}{2}$. □

Note that a cycle $C_n$ of order $n$ can be seen as an $(n, 0)$-tadpole. If $n$ is odd, every vertex $v$ in $C_n$ has two diametral vertices (see Figure 5), ergo $\{v\}$ is a 2-antiresolving set, which leads to the following result.

**Proposition 10.** *[11] A cycle graph $C_n$ of odd order satisfies $(2, 1)$-anonymity.*

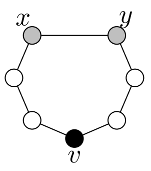

Figure 5: An odd-order cycle containing a vertex $v$ that is equidistant from its two diametral vertices $x$ and $y$.

Based on Theorem 9 and Proposition 10 we provide in Theorems 11 and 12 two sufficient conditions for an edge to be an anonymising edge.

**Theorem 11.** *Let $G = (V, E)$ be a simple and connected graph, $\{v\}$ a 1-antiresolving set of $G$, and $v_1 \dots v_m$ an eccentricity path of $v$ where $v_1 = v$. Let $i$ and $j$ be the smallest and largest positive integers, respectively, such that $v_i$ and $v_j$ are 1-resolvable by $\{v\}$ in $G$. Let $a, b \in \{1, \dots, m\}$ and let $G' = (V, E \cup \{(v_a, v_b)\})$ be the graph resulting from the addition of the edge $(v_a, v_b)$. Every vertex $v_i$ with $i \in \{1, \dots, m\}$ is not 1-resolvable by $\{v\}$ in $G'$ if the following conditions hold:*

*i.* $1 \leq a \leq i - 1$.

*ii.* $a + 2 \leq b \leq m$.

*iii.* $b - a = 2r$ *and* $j - b < r$.

*Proof.* Due to its length, we develop this proof in Appendix A. □

The following result is analogous to Theorem 11 for the cases where the edge addition induces tadpoles with even-order cycles.

**Theorem 12.** *Let $G = (V, E)$ be a simple and connected graph, $\{v\}$ a 1-antiresolving set of $G$, and $v_1 \dots v_m$ an eccentricity path of $v$ where $v_1 = v$. Let $i$ and $j$ be the lowest and largest positive integers, respectively, such that $v_i$ and $v_j$ are 1-resolvable by $\{v\}$ in $G$. Let $a, b \in \{1, \dots, m\}$ and let $G' = (V, E \cup \{(v_a, v_b)\})$ be the graph resulting from the addition of the edge $(v_a, v_b)$. Every vertex $v_i$ with $i \in \{1, \dots, m\}$ is not 1-resolvable by $\{v\}$ in $G'$ if the following conditions hold:*

*i.* $1 \leq a \leq i - 1$.

*ii.* $a + 2 \leq b \leq m$.

*iii. If* $b - a = 2r + 1$, *then* $j - b \leq r \leq m - b$.

*Proof.* Due to its length, we develop this proof in Appendix B. □

The two sufficient conditions provided by Theorems 11 and 12 are used to determine a non-trivial set of anonymising edges, as depicted in Algorithm 2. For every 1-antiresolving set $\{v\}$, Algorithm 2 looks for edges satisfying the sufficient condition stated in either Theorem 11 or Theorem 12. Given an eccentricity path $v_1 \dots v_m$ of the vertex $v = v_1$, finding all such edges can be done in linear time in terms of the size of the eccentricity path, as they can be computed based on $m$ and the smallest and largest positive integers $i$ and $j$, respectively, such that $v_i$ and $v_j$ are 1-resolvable by $v$. In consequence, the time-complexity of Algorithm 2 is $\mathcal{O}(|V|^2)$.

**Algorithm 2**
INPUT: A $(1,1)$-anonymous graph $G=(V,E)$ without end-vertices.
OUTPUT: A set $E'$ of anonymising edges.

```
Let $E'$ be an empty set of edges
for every 1-antiresolving set $\{v\}$ in $G$ do
  Let $v_1 \dots v_m$ be an eccentricity path of $v$, where $v_1 = v$
  Let $i$ and $j$ be the smallest and largest positive integers, respectively, such that $v_i$ and $v_j$ are 1-resolvable by $v$ in $G$.
  for $a = 1$ to $i-1$ do
    for $b = a+2$ to $m$ do
      if $b-a \equiv 0 \pmod 2$ and $j-b < \frac{b-a}{2}$ then
        Add the edge $(v_a, v_b)$ to $E'$
      end if
      if $b-a \equiv 1 \pmod 2$ and $j-b \leq \frac{b-a}{2} \leq m-b$ then
        Add the edge $(v_a, v_b)$ to $E'$
      end if
    end for
  end for
end for
return $E'$
```

## 5 Correctness, convergence, and complexity results

In this section we analyse the correctness, convergence, and complexity of our approach. The fact that Algorithm 2 outputs a set of anonymising edges follows directly from Theorems 11 and 12. The following result, in addition to proving that the edge addition strategy converges to a graph without 1-antiresolving singletons, provides an upper bound on the number of steps in which this is achieved.

**Theorem 13.** *Let $G$ be a simple graph. We define a sequence of graphs $G_i$ (for $i \geq 0$) inductively as follows:*

- *$G_0 = G$.*
- *If there exists a 1-antiresolving set $\{v\}$ in $G_i$, then $G_{i+1}$ is the result of adding an anonymising edge to $G_i$.*
- *Otherwise, $G_{i+1} = G_i$.*

*Let $S_i$ be the set of vertices in $G_i$ such that $v \in S_i$ implies that $\{v\}$ is a 1-antiresolving set in $G_i$. Then $S_i$ is empty for $i \geq \sum_{\forall v \in V} \epsilon_{G_0}(v) - |V| - 1$.*

*Proof.* Consider $G_{i-1} = (V, E_{i-1})$ and $G_i = (V, E_i)$ where $G_{i-1} \neq G_i$. That is, $G_i$ results from adding an anonymising edge to $G_{i-1}$. Let $v_1 \dots v_m$ be an eccentricity path, in $G_{i-1}$, of a vertex $v$ such that $v_1 = v$ and $E_i = E_{i-1} \cup \{(v_a, v_b)\}$ for some $a, b \in \{1, \dots, m\}$. We have that

$$\begin{aligned} d_{G_i}(v_1, v_m) &= d_{G_i}(v_1, v_a) + d_{G_i}(v_a, v_b) + d_{G_i}(v_b, v_m) \\ &= d_{G_{i-1}}(v_1, v_a) + 1 + d_{G_{i-1}}(v_b, v_m). \end{aligned}$$

Since the edge $(v_a, v_b)$ satisfies $d_{G_{i-1}}(v_a, v_b) \geq 2$, we have that $d_{G_{i-1}}(v_1, v_m) > d_{G_i}(v_1, v_m)$, so $\epsilon_{G_i}(v) < \epsilon_{G_{i-1}}(v)$.

The result above states that every edge addition that transforms a graph $G_{i-1}$ into $G_i$ decreases the eccentricity of $v$. Since the length of an eccentricity path cannot be shorter than 1, the maximum number of edge additions that can be applied to $G_0$ for any vertex $v$ is bounded by $\epsilon_{G_0}(v) - 1$. In the worst case, it is possible that every vertex forms a 1-antiresolving set and that the last transformation is the one adding the edge $(u, v)$ that transforms the graph $G_{i-1}$ into a complete graph. Taking into account that in this case both $u$ and $v$ are 1-antiresolving sets of $G_{i-1}$, and they are not 1-antiresolving sets of $G_i$, we obtain the upper bound $\sum_{\forall u \in V} \epsilon_{G_0}(v) - |V| - 1$ for the total number of edge additions that can be applied. Thus, the graph $G_i$ with $i = \sum_{\forall v \in V} \epsilon_{G_0}(v) - |V| - 1$ does not contain 1-antiresolving sets. □

The upper bound provided in Theorem 13 is tight. That is, there exists a graph $G = (V, E)$ such that the number of edges added by our approach equals $\sum_{\forall v \in V} \epsilon_G(v) - |V| - 1$. Moreover, such an upper bound corresponds to the minimum number of edge additions required to transform $G$ into a graph $G'$ that is not $(1, 1)$-anonymous. Such a graph $G$ can be constructed as follows.

Consider the complete graph $K_n = (V, E)$ such that $V = \{v_1, \dots, v_n\}$, $n \geq 3$. Given a vertex $v_{n+1}$, we construct $G = (V', E')$, where $V' = V \cup \{v_{n+1}\}$ and $E' = E \cup \{(v_1, v_{n+1}), (v_2, v_{n+1})\}$ (see Figure 6 for an example). We have that all the edges that can can be added to $G$ have the form $(v_i, v_{n+1})$, for some $i \in \{3, \dots, n\}$. Every such addition makes the distance between $v_i$ and $v_{n+1}$ become 1. Moreover, if any such edge $(v_i, v_{n+1})$ is not added to $G$, then the distance between $v_i$ and $v_{n+1}$ remains equal to 2, causing $v_{n+1}$ to be 1-resolvable by $\{v_i\}$. Therefore, there exists only one way of transforming $G$ into a graph that is not $(1, 1)$-anonymous, that is, transforming it into the complete graph $K_{n+1}$ on $V'$. This requires $n-2$ additional edges, and we have that

$$\begin{aligned} &\sum_{\forall v \in V'} \epsilon_G(v) - |V'| - 1 = \\ &= \epsilon_G(v_1) + \epsilon_G(v_2) + \sum_{v \in \{v_3, \dots, v_n\}} \epsilon_G(v) + \epsilon_G(v_{n+1}) - |V'| - 1 = \\ &= 1 + 1 + 2(n-2) + 2 - (n+1) - 1 = n - 2. \end{aligned}$$

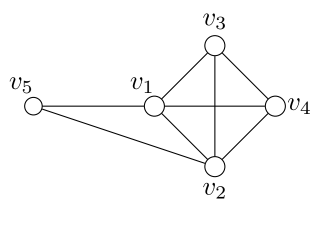

Figure 6: A graph where the upper bound given in Theorem 13 is attained.

Provided with an upper-bound on the number of edges that should be added for the resulting graph to become $(k, \ell)$-anonymous for some $k > 1$ or some $\ell > 1$, we finalise the complexity analysis of our anonymisation approach. Recall that, in our approach, an original graph $G$ is transformed first into an end-vertex-free graph (if the initial graph is not already end-vertex free). Then, we iteratively add anonymising edges until a graph without 1-antiresolving sets is obtained. The number of edge additions depends on how fast these transformations converge to a graph without 1-antiresolving sets. According to Theorem 13, this number is upper-bounded by $\sum_{\forall v \in V} \epsilon_G(v) - |V| - 1$, which is greater than or equal to $|V|(D(G) - 1)$, where $D(G)$ represents the diameter of $G$, that is, the maximum distance between a pair of vertices of $G$. Considering that the time complexities of finding the shortest paths between every pair of vertices in a graph, and that of an execution of Algorithm 2, are $\mathcal{O}(|V|^3)$ and $\mathcal{O}(|V|^2)$, respectively, we obtain that the time complexity of our approach is $\mathcal{O}(|V|^4(D(G) - 1))$. Note that we are assuming that choosing an edge from an anonymising edge set can be done in constant time. That is indeed the case for the three criteria we introduce in the next section for empirical evaluation of our methods in terms of attack resistance and utility preservation. Nevertheless, more elaborate (and thus more time-consuming) selection criteria can also be integrated into the general formulation of our strategy.

## 6 Experiments

In this section we study the proposed anonymisation strategy in terms of privacy and utility preservation. Privacy is measured as the resistance of anonymised graphs to the *walk-based attack* introduced in [1], which is a paradigmatic, purely active attack, in the sense that it requires no externally obtained background information. Utility is measured in terms of the variation of several graph parameters, as well as an application-oriented measure, which assesses the effect of our anonymisation on the outputs of two state-of-the-art community detection algorithms. The experiments were performed on the HPC platform of the University of Luxembourg [22].

### 6.1 Experimental setting

We set up the experiments with three real-life social graphs, $250,000$ synthetic graphs, and three instantiations of the anonymization procedure described in Algorithm 1.

#### 6.1.1 Three perturbation methods

As we mentioned previously, our anonymisation strategy can be instantiated into different methods by varying the criteria used to select the edges to add at every step of the iterative process. We will use these three anonymisation variants to illustrate the fact that, while offering the same level of protection against active attacks, they display different behaviours in terms of utility preservation. We will highlight a number of cases where different utility measures are optimised by different anonymisation variants.

The three variants used for the experiments in this section differ in the criteria applied to select the edge $(v_a, v_b)$ to add at every iteration among those satisfying the conditions of Definition 8. The criteria are the following:

- $(v_a, v_b)$ is the $v$-transformation described in [11]. As this transformation always results in the creation of an odd-order cycle, we will refer to this variant as the *Odd-order cycles variant*, or *OOCV* for short.

- $v_a$ and $v_b$ are as close as possible. We will refer to this variant as the *Smallest-order cycles variant*, or *SOCV* for short. If this condition is satisfied by more than one pair of vertices, we randomly pick one of them.

- $v_a$ and $v_b$ are as distant as possible. We will refer to this variant as the *Largest-order cycles variant*, or *LOCV* for short. As in the previous case, if this condition is satisfied by more than one pair of vertices, we randomly pick one of them.

It is worth remarking that each of these criteria can be enforced directly in Algorithm 2 at no additional computational cost, by keeping track of the globally best anonymising edge found after each step.

### 6.1.2 Three real-life graphs and a large collection of synthetic graphs

Our analyses will be conducted on three real-life social graphs. The first one was obtained from $10$ so-called *ego-networks* in Facebook [12]. An ego-network is the subgraph induced by the set of nodes representing all friends of a given user. The information used to construct the Facebook graph was obtained in 2012 by setting up an app in this social network to which real users could voluntarily sign in and share their friend lists. In addition to the friendship relations, volunteers were instructed to manually classify their friends into categories, referred to as *circles*. A circle may represent, for instance, the set of classmates, work colleagues, family members, etc. The obtained graph contains $4,039$ nodes and $88,234$ relations, as well as membership annotations for $193$ circles. It has diameter $8$ and radius $4$.

The second social graph, which is commonly referred to as the *Panzarasa graph*, after one of its creators [15], was collected from an online community of students at the University of California. The Panzarasa graph contains $59,835$ edges representing messages sent between $1,899$ students from March 23th to October 26th, $2004$. A pair of users is considered to be connected if they exchanged at least one message in either direction. The original graph contains loops, as users were allowed to send messages to themselves, and six isolated vertices. We pre-processed this original graph in order to obtain a simple, undirected and connected graph of order $1,893$ with $20,296$ edges. The Panzarasa graph has diameter $8$ and radius $4$.

Finally, the third network was constructed in 2012 with data from e-mail communications at Universitat Rovira i Virgili (URV), Spain [3]. The authors curated the original data by eliminating group messages with more than 50 recipients, considering that a connection existed between two users if at least one message had been sent in each direction, and eliminating isolated vertices and connected components of order 2. The final graph contains $1,133$ users and $5,451$ relations. It has diameter $8$ and radius $5$.

All three real-life graphs are characterised by very low densities. In order to enrich our analysis on the effectiveness of our methods for counteracting the attack, in Section 6.2.2 we will additionally use a large collection of randomly generated graphs. This collection is composed of $250,000$ graphs for each density value in the set[3] $\{0.03, 0.04, \ldots, 1\}$. We fixed $100$ as the number of vertices in each graph, and each edge set is created by connecting random pairs of vertices, until the number of edges corresponding to the desired density value is reached.

[3]We start with the density value 0.03 because there exist no connected graphs of order 100 and density 0.01, whereas for the density value 0.02 the only connected graph of order 100 is the path graph $P_{100}$.

## 6.2 Privacy analysis

We will first conduct an analysis on the resistance of our methods to the walk-based attack. As discussed above, our focus will be on showing that the three variants used to showcase our general strategy provide the same level of protection against this attack. We will first describe the attack in detail. Then, we will describe the manner in which we compute the adversary's probability of success. Finally, we will discuss the obtained results.

### 6.2.1 Simulation of the walk-based attack

Given a social graph $G = (V, E)$, the walk-based attack targets a set $Y = \{y_1, \dots, y_m\}$ of users in $G$. Before the publication of the graph, the attacker performs the following actions, as described in [1]:

1. Insert a set of sybil vertices $X = \{x_1, \dots, x_n\}$ into $G$.
2. Map each vertex $y_i \in Y$ to a subset $N_i \subseteq X$ of sybil vertices, called the *fingerprint* of $y_i$, by connecting $y_i$ to all vertices in $N_i$. The mapping function is made injective to guarantee that all fingerprints are pairwise different.
3. Create an internal connection pattern between vertices in $X$, which is crafted to allow efficient recovery of the subgraph induced by $X$ in the released graph. To that end, all edges $(x_i, x_{i+1})$, $1 \le i \le n-1$, are added deterministically and every other pair is added with probability $1/2$.

We will denote the graph obtained after simulating this phase of the attack by $G' = (V', E')$, where $V' = V \cup X$ and $E' = E \cup F$, with $F$ composed of all edges added in steps 2 and 3 above. In addition, we use $G''$ to denote the anonymisation of $G'$, and $G'(X)$ (resp. $G''(X)$) to denote the subgraph of $G'$ (resp. $G''$) induced by the vertices in $X$. Note that, $G'' = G'$ when no anonymisation is considered, as in the original article by Backstrom et al. [1].

In the second phase of the attack, once $G''$ is published, the adversary looks for the subgraph $G'(X)$ within $G''$, by performing a greedy search using the sequence of degrees of the sybil nodes. According to Backstrom et al. [1], this phase of the attack may fail in several ways, namely (*i*) there exist other subgraphs isomorphic to $G'(X)$, (*ii*) $G'(X)$ has non-trivial automorphisms, or (*iii*) the anonymised version $G''$ of $G'$ no longer contains the subgraph $G'(X)$. We account for these three cases in the computation of the success probability, and proceed as follows:

1. *Quantifying probability of success for the retrieval phase:* We compute the set $\mathcal{X}$ containing all isomorphisms from subgraphs of $G''$ to $G'(X)$. If $|\mathcal{X}| = 0$ then the attack fails, as the adversary is unable to retrieve the planted graph. On the contrary, if $|\mathcal{X}| = 1$ then this phase of the attack succeeds. Finally, in the case where $|\mathcal{X}| > 1$, the adversary can still guess the correct vector of sybil nodes with probability $1/|\mathcal{X}|$. For details on how this probability grows as the cardinality of $X$ increases, we refer the reader to [1].
2. *Quantifying probability of success for the re-identification phase:* Let $\hat{X} = \{\hat{x}_1, \dots, \hat{x}_n\} \subseteq V'$ be the set of vertices that the adversary has mapped to $X$ after the retrieval phase. We now determine, for each fingerprint $N_i \subseteq X$, $i \in \{1, \dots, m\}$, the candidate set $V_i = \{y' \in V' \mid \forall x_j \in N_i : (y', \hat{x}_j) \in E''\}$ containing all vertices in $V'$ whose fingerprint according to $\hat{X}$ is determined by $N_i$. Again, prior to anonymisation, each candidate

set $V_i$, $i \in \{1, \dots, m\}$, is highly likely to have cardinality $1$, which would allow the adversary to uniquely re-identify the targeted victim $y_i$. This is not necessarily the case after anonymisation, and we assume that $y_i$ can be mapped to any member of $V_i$ with equal probability, so $y_i$ is correctly identified with probability $1/|V_i|$.

3. *Computing the joint probability:* We consider that the adversary succeeds if all vertices in $Y$ are correctly re-identified. Therefore, the probability of success of the walk-based attack is $0$ if $|\mathcal{X}| = 0$, otherwise it is computed by the following formula:

$$\frac{\sum_{G(X)\in\mathcal{X}} \prod_{1\leq i\leq m} p_i}{|\mathcal{X}|} \quad \text{where} \quad p_i = \begin{cases} 1/|V_i| & \text{if} \quad y_i \in V_i \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

### 6.2.2 Evaluation of attack resistance

We first proceed to assess to what extent the theoretical privacy guarantee proven for our method translates into resistance to active attackers with the ability to insert one sybil node in the graph.

We will first focus on the three real-life graphs. We simulate an attack on each graph by adding one sybil node and linking it to a randomly selected vertex of the graph. Then, for every graph we obtain three anonymised versions, one corresponding to each variant of our strategy, and compute the success probabilities of the attack on these anonymised graphs. This process was repeated $50$ times for each graph and anonymisation method. The averaged results are shown in Table 1. In the table, we show the average success probability of the attack on the original graph (identified as *Orig*). Additionally, for each graph and anonymisation method, we show the following two average success probability values:

- *Ours*: for each run, the success probability of each attack is computed on the anonymised graph obtained as a result of applying the corresponding variant of our strategy.
- *Rand*: for each run, the success probability of each attack is computed on a perturbed graph obtained by randomly inserting as many edges as those inserted by the corresponding variant of our strategy.

| Variant | Facebook | | | Panzarasa | | | URV | | |
|---|---|---|---|---|---|---|---|---|---|
| | Orig | Ours | Rand | Orig | Ours | Rand | Orig | Ours | Rand |
| Odd-ord. cyc. | | 0.0 | 0.0123 | | 0.0 | 0.0009 | | 0.0 | 0.0017 |
| Small.-ord. cyc. | 0.1631 | 0.0 | 0.0110 | 0.1657 | 0.0 | 0.0011 | 0.1786 | 0.0 | 0.0035 |
| Larg.-ord. cyc. | | 0.0 | 0.0126 | | 0.0 | 0.0016 | | 0.0 | 0.0044 |

Table 1: Average success probabilities of the walk-based attack on the Facebook, Panzarasa and URV graphs.

As can be observed in the table, the success probability of the attack on every graph after anonymisation was zero for all three variants of our anonymisation approach. This value is lower than the theoretical upper bound because of a side-effect of the anonymisation algorithm, which makes several of the added edges be incident in sybil nodes. This not only contributes to changing the fingerprints of the victims, but also hampers the attacker's ability to retrieve the sybil nodes. That is to say, the cardinality of the set $\mathcal{X}$ becomes zero after anonymisation. This phenomenon also manifests itself when edges are randomly added, causing the success probability of the attack to also be low in these cases. Note that the success probabilities are considerably larger for the original graph in all three cases, which shows that the original graphs were indeed more vulnerable to the attack. The reason why these probabilities are not close to 1 is that several legitimate vertices in these graphs, namely those having degree 1, are ambiguously retrievable along with the inserted sybil.

The three real-life graphs have in common the fact that their densities are considerably small. However, it has been observed that social network graphs tend to become more dense as they evolve [8]. Considering this, we will use the collection of randomly generated graphs described in Section 6.1.2 to analyse the behaviour of our methods in a more diversified setting in terms of density. For every randomly generated graph, we applied the same steps as for the three real-life networks for simulating the attack, obtaining the anonymised versions and computing the success probability of the attack. Finally, these probabilities were averaged over the $250,000$ graphs generated for each density value. The results are depicted in Figure 7.

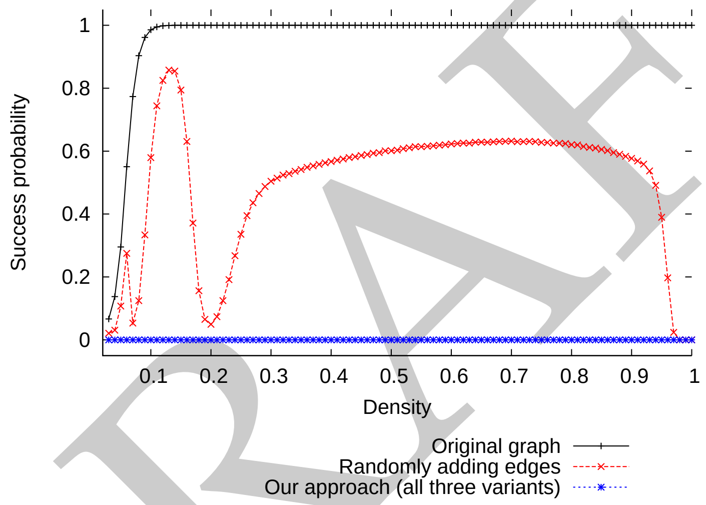

Figure 7: Average success probabilities of the walk-based attack with one sybil node.

As depicted in the figure, the success probability of the attack on every graph after anonymisation was zero for all three variants of our anonymisation approach and all density values. This observation is important, because it shows that the behaviour previously observed in Table 1 is not constrained to these three networks, and indeed it holds for all density values. Additionally, the figure highlights the pertinence of applying our anonymisation methods instead of simply introducing an equivalent amount of random perturbations in the graph structure. As can be observed, for most density values, the attacker would perform much better on the randomly perturbed graphs than on the ones anonymised by our methods. Finally, the figure shows that the original graphs are in all cases more vulnerable to the attacks than the anonymised versions, and the success probability is close to 1 in most cases. As observed for the Facebook, Panzarasa and URV graphs, for the graphs with the lowest density values this probability is not so close to 1 because of the ambiguity in retrieving the inserted sybil.

To conclude our analysis, we now turn to assessing whether, and to what extent, our anonymisation strategy is able to provide some level of protection against the attack beyond the theoretical guarantee, in particular when facing attackers with the ability to leverage more than one sybil node. Figure 8 shows the averaged success probability of the attack with two and four sybil nodes. For density values above $0.05$, the three variants of our strategy exhibit almost identical behaviour in terms of attack resistance, but for lower density values the Largest-order cycles variant shows a higher resistance to the attacks that introduce several sybil nodes. The reason for this greater resistance is that, when this variant is applied, the likelihood of the added edges to be incident on some of the sybil nodes is higher. Thus, this variant is more likely to prevent the adversary from successfully retrieving the attacker subgraph. For each density value, Figure 8 shows the success probability of the attack after applying the Largest-order cycles variant, along with the success probability of the attack when randomly inserting as many edges as this variant. As the figure shows, the success probabilities of the attack with two and four sybil nodes on the graph anonymised by our method remain considerably low for most density values. Moreover, in these settings our method still significantly outperforms random perturbation for most density values.

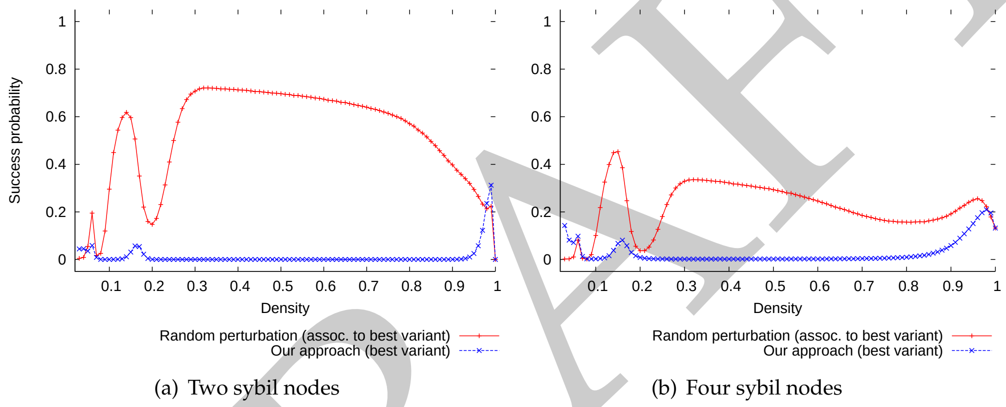

Figure 8: Average success probabilities of the walk-based attack with (a) two and (b) four sybil nodes.

## 6.3 Utility analysis

As we mentioned previously, our utility analysis will focus on the variation of several graph parameters, as well as an application-oriented measure, which assesses the effect of our anonymisation on the outputs of two state-of-the-art community detection algorithms.

### 6.3.1 Analysis of graph parameters

Tables 2, 3, 4 and 5 show the extent to which some graph parameters changed after anonymisation for each variant and graph. In the tables, $G$ represents the original graph and $G'$ an anonymised version. Table 2 shows the number of edges added by each variant, whereas Table 3 focuses on the variations of diameter, effective diameter and radius. Recall that, in a graph $G$, the diameter $D(G)$ is the maximum distance between a pair of vertices of $G$, or in other words the maximum eccentricity value of a vertex of $G$, whereas the radius $R(G)$ is the minimum eccentricity value, i.e. $D(G) = \max_{v\in V(G)}\{\epsilon_G(v)\}$ and $R(G) = \min_{v\in V(G)}\{\epsilon_G(v)\}$. The effective diameter is defined in [7] as the minimum number of hops in which $90\%$ of all connected pairs of vertices can reach each other, having the effect of mitigating the influence of exceptionally large distance values on the estimation of distances within the social graph. Finally, Tables 4 and 5 depict the effect of the anonymisation on other commonly used global statistics of the graphs. The tables show the cosine similarities between the degree distributions and the variations in the clustering coefficient, which is defined as the proportion of triangles among the vertex triples $(v_i, v_j, v_k)$ such that $v_i \sim v_j \sim v_k$.

| Variant | $\|E(G')\| - \|E(G)\|$ | | |
|---|---|---|---|
| | Facebook | Panzarasa | URV |
| Odd-order cycles | 74 (+0.08%) | 405 (+2.00%) | 244 (+4.48%) |
| Smallest-order cycles | 73 (+0.08%) | 417 (+2.05%) | 204 (+3.74%) |
| Largest-order cycles | 73 (+0.08%) | 478 (+2.36%) | 306 (+5.61%) |

Table 2: Number of added edges after applying each anonymisation variant to the Facebook, Panzarasa and URV graphs. The parenthesised percentages denote the relative growth of the number of edges due to our anonymisation algorithm.

The most relevant conclusion that we extract from the results shown in Table 2 is that, even if the (considerably large) theoretical upper bound for the number of added edges given in Theorem 13 is sharp, considerably fewer edge additions are needed in real-life graphs. An interesting observation obtained from the analysis of both tables is that the Facebook graph, which is the largest in terms of order and number of edges, is the one that suffers the smallest amount of modification. Arguably, this may be the result of obfuscating subgraphs being more likely to naturally occur in this social graph. Moreover, we note that the three variants of our anonymisation strategy display the same behaviour in terms of variation in diameter, effective diameter and radius, as shown in Table 3. It is worth noting that, while the variations in diameter observed for the Panzarasa and URV graphs may seem considerably large in light of the small world notion, they are mostly due to the shortening of exceptionally long paths, as suggested by the comparatively smaller variations in radius and effective diameter.

| Variant | $D(G') - D(G)$ | | | $ED(G') - ED(G)$ | | | $R(G') - R(G)$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | FB | Panz. | URV | FB | Panz. | URV | FB | Panz. | URV |
| Odd-ord. cyc. | 0 | -3 | -2 | 0 | 0 | -1 | 0 | -1 | -1 |
| Smallest-ord. cyc. | 0 | -3 | -2 | 0 | 0 | -1 | 0 | -1 | -1 |
| Largest-ord. cyc. | 0 | -3 | -2 | 0 | 0 | -1 | 0 | -1 | -1 |

Table 3: Variations in diameter, effective diameter and radius after applying each anonymisation variant to the Facebook, Panzarasa and URV graphs.

Finally, we note that global utility indicators, such as the degree distribution and the clustering coefficient, suffer very small variations, as shown in Tables 4 and 5, which is a positive observation, as it implies that the changes introduced by the anonymisation methods had a limited global impact on the network structure.

| Variant | *DD cosine similarity* | | |
|---|---|---|---|
| | FB | Panz. | URV |
| Odd-order cycles | 0.9999 | 0.9998 | 0.9991 |
| Smallest-order cycles | 0.9999 | 0.9998 | 0.9992 |
| Largest-order cycles | 0.9999 | 0.9997 | 0.9988 |

Table 4: Cosine similarities between degree distributions after applying each anonymisation variant to the Facebook, Panzarasa and URV graphs.

| Variant | $CC(G') - CC(G)$ | | |
|---|---|---|---|
| | FB | Panz. | URV |
| Odd-ord. cyc. | $-1.221 \times 10^{-4}$ (-0.02%) | $1.305 \times 10^{-4}$ (+0.09%) | $-2.173 \times 10^{-3}$ (-0.58%) |
| Smallest-ord. cyc. | $-6,672 \times 10^{-5}$ (-0.01%) | $4.152 \times 10^{-4}$ (+0.27%) | $1.951 \times 10^{-4}$ (+0.05%) |
| Largest-ord. cyc. | $-9.646 \times 10^{-5}$ (-0.01%) | $-1.177 \times 10^{-3}$ (-0.79%) | $-5.429 \times 10^{-3}$ (-1.45%) |

Table 5: Variations in clustering coefficients after applying each anonymisation variant to the Facebook, Panzarasa and URV graphs.

### 6.3.2 Community detection analysis

Now, we turn to assessing the impact of the perturbations introduced by our methods on application-oriented utility, for which we focus on a popular network analysis task: community detection. As we mentioned previously, the purpose of community detection is to find, and group together, sets of vertices that show some common behaviour. This behaviour may be given by the way in which they connect to each other and to other vertices of the network. While, by definition, communities may be distinguishable as a whole, this does not necessarily imply that the identity of the vertices in a community is deducible from the community itself. From the perspective of anonymisation, it is thus desirable that a method produces graphs where, even though individual vertices are no longer identifiable, the community structure changes as little as possible.

We used two state-of-the-art algorithms[4]: BigClam [24] and CoDA [25]. Both algorithms were designed for efficiently detecting communities in large scale social graphs, relying only on the graph structure. This aspect makes them suitable for our experiments, as we handle graphs lacking vertex identifiers and attributes. According to the authors of [25], experiments on the Facebook graph described above showed that the communities found by CoDA are the ones that better match the sets of circles defined by the users. In these experiments, BigClam ranked second, only outperformed by CoDA.

Let $G = (V, E)$ be a social graph and let $\mathcal{Q} = \{Q_1, \dots, Q_p\}$, $Q_i \in V$ for $i \in \{1, \dots, p\}$, be the set of communities of $G$, manually annotated by a human expert. The set $\mathcal{Q}$ is usually referred to as the *gold standard*. Now, let $\mathcal{C} = \{C_1, \dots, C_q\}$, $C_i \in V$ for $i \in \{1, \dots, q\}$, be the output of a community detection algorithm on $G$. In order to assess the quality of this output, we follow the methodology applied in [7, 24, 25]. First, every pair of communities

[4]We used the implementations included in the library SNAP 3.0 [9], developed within the *Stanford Network Analysis Project*. For additional information on SNAP visit `http://snap.stanford.edu` and for a survey on community detection algorithms see [16].

$Q_i \in \mathcal{Q}, C_j \in \mathcal{C}$ is compared using the $F_1$-score [21], which is defined as follows:

$$F_1(Q_i, C_j) = \frac{2 \cdot precision(Q_i, C_j) \cdot recall(Q_i, C_j)}{precision(Q_i, C_j) + recall(Q_i, C_j)} \tag{2}$$

where

$$precision(Q_i, C_j) = \frac{|Q_i \cap C_j|}{|C_j|} \tag{3}$$

and

$$recall(Q_i, C_j) = \frac{|Q_i \cap C_j|}{|Q_i|} \tag{4}$$

The measures defined by Equations 2, 3 and 4 were introduced in the field of Information Retrieval. The gold standard community $Q_i$ is assumed to be the set of objects relevant to some query, that is the set of objects that an ideal algorithm should retrieve, whereas $C_j$ is viewed as the set of objects retrieved by the algorithm being evaluated. Ideally, $C_j$ should be equal to $Q_i$. Thus, $precision$ determines the ratio of retrieved objects that are relevant, whereas $recall$ determines the ratio of relevant objects that the algorithm is able to retrieve. The $F_1$-score coincides with the double of the harmonic mean of $precision$ and $recall$, and it ranges in the interval $[0, 1]$. Note that if $C_j = Q_i$, then $F_1(Q_i, C_j) = 1$. In the community detection scenario, pairwise $F_1$ values between communities are combined into a global score as follows:

$$score(\mathcal{Q}, \mathcal{C}) = \frac{1}{2|\mathcal{Q}|} \sum_{i=1}^{p} \max_{j \in \{1,\dots,q\}} \{F_1(Q_i, C_j)\} + \frac{1}{2|\mathcal{C}|} \sum_{j=1}^{q} \max_{i \in \{1,\dots,p\}} \{F_1(Q_i, C_j)\} \tag{5}$$

That is, every gold standard community is matched to its most similar community in the output of the algorithm, and *vice versa*, and the $F_1$-scores for all pairs are averaged. We will use this quality measure as the basis for utility evaluation as follows. Let $G = (V, E)$ be a social graph and $\mathcal{Q} = \{Q_1, \dots, Q_p\}$, $Q_i \in V$ for $i \in \{1, \dots, p\}$, the gold standard set of communities of $G$. Let $G' = (V, E')$ be an anonymised version of $G$ and let $\mathcal{C} = \{C_1, \dots, C_q\}$, $C_i \in V$ for $i \in \{1, \dots, q\}$, and $\mathcal{C}' = \{C'_1, \dots, C'_r\}$, $C'_i \in V$ for $i \in \{1, \dots, r\}$, be the outputs of a community detection algorithm on $G$ and $G'$, respectively. Our utility measure seeks to assess to what extent the output of the community detection algorithm is perturbed by the anonymisation, and is determined by the following loss function:

$$loss(G, G') = \frac{|score(\mathcal{Q}, \mathcal{C}) - score(\mathcal{Q}, \mathcal{C}')|}{score(\mathcal{Q}, \mathcal{C})} \tag{6}$$

Out of the three benchmark social graphs that we described above, we will now focus on the Facebook graph, since this is the one for which gold standard community annotation is available. In a manner analogous to the one described in [25], we will consider a community detection problem for each ego-network, consisting of finding a set of communities that are as similar as possible to the set of circles labelled by the ego-network owner. We applied the three variants of our anonymisation strategy to the subgraphs corresponding to each ego-network, and compared the outputs of BigClam and CoDA on each anonymised version to the outputs on the original subgraphs, to obtain the utility loss according to Equation 6. Table 6 shows the obtained results, rounded up to four significant figures.

From the analysis of these results, the most notable observation is that, on average, all three variants cause very small utility losses, which means that they do not considerably perturb the performance of the community detection algorithms on the anonymised

| Ego-network id | BigClam | | | CoDA | | |
|---|---|---|---|---|---|---|
| | OOCV | SOCV | LOCV | OOCV | SOCV | LOCV |
| 0 | 0.1646 | 0.1099 | 0.0281 | 0.0938 | 0.0013 | 0.0487 |
| 107 | 0.0318 | 0.0548 | 0.0195 | 0.0206 | 0.0238 | 0.0427 |
| 348 | 0.2148 | 0.0322 | 0.1161 | 0.0481 | 0.0289 | 0.0257 |
| 414 | 0.1136 | 0.0727 | 0.0538 | 0.0388 | 0.0073 | 0.0721 |
| 686 | 0.0294 | 0.0763 | 0.0760 | 0.0047 | 0.0436 | 0.0134 |
| 698 | 0.0303 | 0.1988 | 0.1595 | 0.0993 | 0.1950 | 0.0038 |
| 1684 | 0.0728 | 0.0318 | 0.0084 | 0.0334 | 0.0347 | 0.0260 |
| 1912 | 0.0363 | 0.0023 | 0.0134 | 0.0297 | 0.0284 | 0.0061 |
| 3437 | 0.0794 | 0.1730 | 0.0484 | 0.0050 | 0.0190 | 0.0196 |
| 3980 | 0.1719 | 0.0751 | 0.1329 | 0.0122 | 0.1102 | 0.0185 |
| Average | 0.0772 | 0.0827 | 0.0656 | 0.0386 | 0.0492 | 0.0277 |

Table 6: Utility loss on the ego-networks of the Facebook graph, applying BigClam and CoDA community detection.

graphs. A more fine-grained analysis shows no clear trend of some variant of the anonymisation strategy considerably outperforming the others in terms of utility. Instead, we see that for each variant there is some ego-network and some community detection algorithm for which it obtains the best utility score. On average, the Largest-order cycles variant performs slightly better, followed by OOCV and SOCV. We conjecture that the reason for this global trend may be that adding edges between originally distant vertices introduces smaller amounts of noise in the models used by the community detection algorithms.

Summing up the results in this section, we corroborated in Section 6.2 that all three instances of our anonymisation strategy effectively counteract active attacks when the adversary has the ability to insert one sybil node in the graph. Now, we have seen how the flexibility of our proposal, along with the invariant theoretical privacy guarantee, may allow the practitioner to focus on optimising the desired utility measure depending on the input graph. This process needs to be conducted in a case-by-case basis, since different utility measures may be optimised by different anonymisation variants.

## 7 Conclusions

In this article we have dealt with the anonymisation of social graphs facing active attacks. We have presented a general edge addition approach which guarantees that the output satisfies $(k, \ell)$-anonymity for some $k > 1$ or some $\ell > 1$. We have proven the theoretical soundness of the approach, its convergence, and given a sharp upper bound on the number of necessary edge additions. Experiments on a large collection of randomly generated graphs for three variants of our strategy show that all three effectively counteract active attacks when the adversary is able to insert one sybil node in the network and, additionally, provide a certain degree of protection when more than one sybil node can be inserted. We compared these three variants in terms of graph-specific and application-oriented utility measures on real-life social graphs. These experiments showed that the number of edge additions needed in real-life graphs is considerably smaller than the theoretical upper bound. Moreover, we found that all three variants cause small variations in global utility measures, as well as small degradations in the outputs of state-of-the-art community detection algorithms. Finally, the results of these experiments also allowed us to to corroborate the fact that our general strategy provides a high degree of flexibility that allows to target utility criteria in a case-specific manner while keeping the privacy guarantees unchanged.

One interesting side-effect of our anonymisation strategy that we observed is the fact that, while it aims to make pairs of vertices undistinguishable by forcing their fingerprints to be identical, it also thwarts the attacks by preventing the adversary from unambiguously retrieving the sybil nodes. This observation suggests two attractive directions for future work. First, it would be interesting to enunciate privacy measures that go beyond $(k, \ell)$-anonymity by additionally accounting for the ambiguity in retrieving the sybil nodes. Secondly, based on these privacy measures, it will be useful to devise anonymisation methods that purposely seek to thwart the sybil retrieval phase and determine which strategy is less expensive in terms of utility loss: (*i*) preventing unambiguous sybil node retrieval, (*ii*) preventing fingerprint-based re-identification, or (*iii*) a combination of both.

## References

[1] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web*, pp. 181–190, New York, NY, USA, 2007. ACM.

[2] Sean Chester, Bruce M. Kapron, Ganesh Ramesh, Gautam Srivastava, Alex Thomo, and S. Venkatesh. k-anonymization of social networks by vertex addition. In *Proceedings II of the 15th East-European Conference on Advances in Databases and Information Systems*, pp. 107–116, Vienna, Austria, 2011.

[3] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt, and Alex Arenas. Self-similar community structure in a network of human interactions. *Physical review E*, 68(6):065103, 2003.

[4] Frank Harary and Robert A. Melter. On the metric dimension of a graph. *Ars Combinatoria*, 2:191–1995, 1976.

[5] Sepp Hartung, André Nichterlein, Rolf Niedermeier, and Ondrej Suchý. A refined complexity analysis of degree anonymization in graphs. *Information and Computation*, 243:249–262, 2015.

[6] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment*, 1(1):102–114, 2008.

[7] Shouling Ji, Weiqing Li, Prateek Mittal, Xin Hu, and Raheem Beyah. Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization. In *Proceedings of the 24th USENIX Security Symposium*, pp. 303–318, Washington DC, USA, 2015.

[8] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 177–187, Chicago, IL, USA, 2005.

[9] Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

[10] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 93–106, Vancouver, Canada, 2008.

[11] Sjouke Mauw, Rolando Trujillo-Rasua, and Bochuan Xuan. Counteracting active attacks in social network graphs. In *Proceedings of DBSec 2016*, volume 9766 of *LNCS*, pp. 233–248, 2016.

[12] Julian Mcauley and Jure Leskovec. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data*, 8(1):4, 2014.

[13] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 223–228, New York, NY, USA, 2004.

[14] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 173–187, Washington, DC, USA, 2009.

[15] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the Association for Information Science and Technology*, 60(5):911–932, 2009.

[16] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

[17] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[18] Peter J. Slater. Leaves of trees. *Congressus Numerantium*, 14:549559, 1975.

[19] Latanya Sweeney. Uniqueness of simple demographics in the U.S. population. Technical report, Carnegie Mellon University, School of Computer Science, Data Privacy Laboratory, 2002.

[20] Rolando Trujillo-Rasua and Ismael González Yero. k-metric antidimension: A privacy measure for social graphs. *Information Sciences*, 328:403–417, 2016.

[21] C. J. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[22] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos. Management of an academic HPC cluster: The UL experience. In *Proceedings of the 2014 International Conference on High Performance Computing & Simulation*, pp. 959–967, Bologna, Italy, 2014.

[23] Peng Wei, Feng Li, Xukai Zou, and Jie Wu. A two-stage deanonymization attack against anonymized social networks. *IEEE Transactions on Computers*, 63(2):290–303, 2014.

[24] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pp. 587–596, Rome, Italy, 2013.

[25] Jaewon Yang, Julian McAuley, and Jure Leskovec. Detecting cohesive and 2-mode communities indirected and undirected networks. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 323–332, New York, NY, USA, 2014.

[26] Haifeng Yu. Sybil defenses via social networks: A tutorial and survey. *SIGACT News*, 42(3):80–101, 2011.

[27] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pp. 3–17, Oakland, CA, USA, 2008.

[28] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 267–278, Pisa, Italy, 2006.

[29] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the IEEE 24th International Conference on Data Engineering*, pp. 506–515, Washington, DC, USA, 2008.

[30] Lei Zou, Lei Chen, and M. Tamer Özsu. K-automorphism: A general framework for privacy preserving network publication. *Proceedings of the VLDB Endowment*, 2(1):946–957, 2009.

## A Proof of Theorem 11

**Theorem 11.** *Let $G = (V, E)$ be a simple and connected graph, $\{v\}$ a 1-antiresolving set of G, and $v_1 \dots v_m$ an eccentricity path of $v$ where $v_1 = v$. Let $i$ and $j$ be the smallest and largest positive integers, respectively, such that $v_i$ and $v_j$ are 1-resolvable by $\{v\}$ in $G$. Let $a, b \in \{1, \dots, m\}$ and let $G' = (V, E \cup \{(v_a, v_b)\})$ be the graph resulting from the addition of the edge $(v_a, v_b)$. Every vertex $v_i$ with $i \in \{1, \dots, m\}$ is not 1-resolvable by $\{v\}$ in $G'$ if the following conditions hold:*

*i.* $1 \leq a \leq i - 1$.

*ii.* $a + 2 \leq b \leq m$.

*iii.* $b - a = 2r$ *and* $j - b < r$.

*Proof.* We already have that $b - a = 2r$ and $j - b < r$. We thus further differentiate two subcases (see Figure 9): Case 1, where $j - r < b < j$, and Case 2, where $j \leq b \leq m$. Note that, in Case 1, the subgraph of $G'$ induced by the set $\{v_a, v_{a+1}, \dots, v_j\}$ is a $(2r + 1, j - b)$-tadpole, whereas in Case 2 the subgraph of $G'$ induced by the set $\{v_a, v_{a+1}, \dots, v_b\}$ is a cycle of order $2r + 1$. We will rely on those facts to prove that a vertex $w \in \{v_1, \dots, v_m\}$ is not 1-resolvable by $\{v\}$ in $G'$.

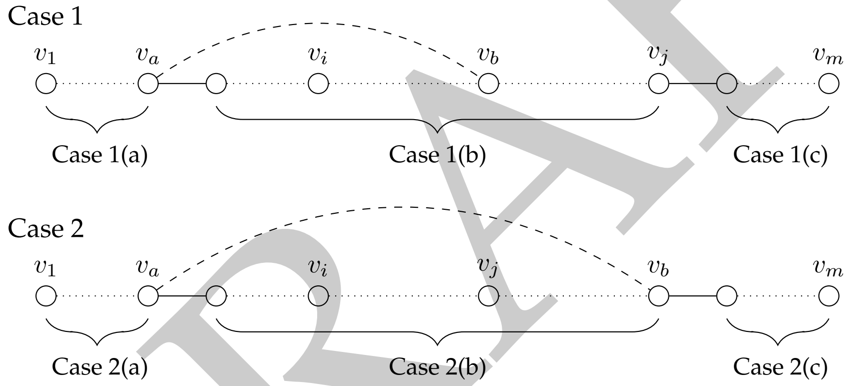

Figure 9: An eccentricity path $v_1 \dots v_m$ within the graph $G$. In both cases, the dashed line represents the edge added to obtain the graph $G'$.

Case 1. ($j - r < b < j$). In this case, we will analyse the following three scenarios regarding the position of the vertex $w$ in the eccentricity path $v_1 \dots v_m$ (see Figure 9, Case 1):

(a) $w \in \{v_1, \dots, v_a\}$. In this case $w$ is not 1-resolvable by $\{v_1\}$ in $G$. Therefore, there exists $w' \in V \setminus \{v_1, \dots, v_m\}$ such that $d_G(v_1, w) = d_G(v_1, w')$. We choose $k \in \{1, \dots, m\}$ to be the largest positive integer such that $d_G(v_1, w') = d_G(v_1, v_k) + d_G(v_k, w')$. On the one hand, it holds that $d_G(v_k, w') = d_{G'}(v_k, w')$. On the other hand, it is easy to note that $k < a$, as otherwise $d_G(v_1, w') \geq a > d_G(v_1, w)$. This implies that $d_G(v_1, v_k) = d_{G'}(v_1, v_k)$ and thus, $d_G(v_1, w) = d_{G'}(v_1, w) = d_G(v_1, w') = d_{G'}(v_1, w')$. In consequence, $w$ is not 1-resolvable by $\{v\}$ in $G'$.

(b) $w \in \{v_{a+1}, \dots, v_j\}$. Let $H$ be the subgraph of $G'$ induced by $\{v_a, v_{a+1}, \dots, v_j\}$. As we pointed out above, $H$ is a $(2r+1, j-b)$-tadpole. We have that the vertex

$v_b$ is the sole degree-3 vertex of $H$. Moreover, $j - b \leq r - 1 = \frac{(2r+1)-3}{2}$, so $\{v_a\}$ is a 2-antiresolving set of $H$ by Theorem 9. In consequence, there exists a vertex $w' \in \{v_{a+1}, \ldots, v_j\} - \{w\}$ such that $d_H(v_a, w) = d_H(v_a, w')$. Since $d_{G'}(v, x) = d_{G'}(v, v_a) + d_{G'}(v_a, x) = a - 1 + d_H(v_a, x)$ for every $x \in V(H)$, we can conclude that $d_{G'}(v, w) = a-1+d_H(v_a, w) = a-1+d_H(v_a, w') = d_{G'}(v, w')$, so $w$ is not 1-resolvable by $\{v\}$ in $G'$.

(c) $w \in \{v_{j+1}, \ldots, v_m\}$ (this case occurs only if $j < m$). Here, as in Case 1(a), since $w$ is not 1-resolvable by $\{v\}$ in $G$, there exists $w' \in V \setminus \{v_1, \ldots, v_m\}$ such that $d_G(v, w) = d_G(v, w')$. Again, let $k$ be the largest positive integer such that $d_G(v, w') = d_G(v, v_k) + d_G(v_k, w')$. We have that $k \geq j$, as $k < j$ contradicts the fact that $v_j$ is 1-resolvable by $\{v\}$ in $G$, so $d_{G'}(v, w) = d_{G'}(v, v_k) + d_{G'}(v_k, w) = d_G(v, v_k) - (b - a) + 1 + d_G(v_k, w) = d_G(v, w) - (b - a) + 1$ and, analogously, $d_{G'}(v, w') = d_G(v, w') - (b - a) + 1$. In consequence, $d_{G'}(v, w) = d_{G'}(v, w')$, so $w$ is not 1-resolvable by $\{v\}$ in $G'$.

Case 2. ($j \leq b \leq m$). In this case, we will analyse the following three scenarios regarding the position of the vertex $w$ in the eccentricity path $v_1 \ldots v_m$ (see Figure 9, Case 2):

(a) $w \in \{v_1, \ldots, v_a\}$. This case is equivalent to Case 1(a). Thus, by analogy, we have that $w$ is not 1-resolvable by $\{v\}$ in $G'$.

(b) $w \in \{v_{a+1}, \ldots, v_b\}$. Let $H$ be the subgraph of $G'$ induced by $\{v_a, v_{a+1}, \ldots, v_b\}$. As we pointed out above, $H$ is a cycle of order $2r+1$, so $\{v_a\}$ is a 2-antiresolving set of $H$ by Proposition 10. In a manner analogous to Case 1(b), there exists a vertex $w' \in \{v_{a+1}, \ldots, v_b\} \setminus \{w\}$ such that $d_{G'}(v, w) = a - 1 + d_H(v_a, w) = a - 1 + d_H(v_a, w') = d_{G'}(v, w')$, so $w$ is not 1-resolvable by $\{v\}$ in $G'$.

(c) $w \in \{v_{b+1}, \ldots, v_m\}$ (this case occurs only if $b < m$). Here, as in Case 1(c), since $w$ is not 1-resolvable by $\{v\}$ in $G$, there exists $w' \in V - \{v_1, \ldots, v_m\}$ such that $d_G(v, w) = d_G(v, w')$. Again, let $k$ be the largest positive integer such that $d_G(v, w') = d_G(v, v_k) + d_G(v_k, w')$. If $k \geq b$ then, as in Case 1(c), $d_{G'}(v, w) = d_G(v, w) + (b - a) + 1 = d_G(v, w') + (b - a) + 1 = d_{G'}(v, w')$, so $w$ is not 1-resolvable by $\{v\}$ in $G'$.
Now, if $b - r < k < b$, then $d_{G'}(v, w) = d_G(v, v_a) + 1 + d_G(v_b, w)$, whereas

$$d_{G'}(v, w') = d_G(v, v_a) + 1 + d_G(v_b, v_k) + d_G(v_k, w') =$$

$$= d_G(v, v_a) + 1 + 2 \times d_G(v_b, v_k) + d_G(v_b, w).$$

As $d_{G'}(v, w) < d_{G'}(v, w')$, we can conclude that there exists a vertex $w''$ in a shortest $v_k - w'$ path such that $d_{G'}(v, w) = d_{G'}(v, w'')$, so $w$ is not 1-resolvable by $\{v\}$ in $G'$.
Finally, if $j \leq k \leq b - r$, we have that $d_{G'}(v, w') = d_{G'}(v, v_a) + d_{G'}(v_a, v_k) + d_{G'}(v_k, w') = d_G(v, v_a)+d_G(v_a, v_k)+d_G(v_k, w') = d_G(v, w')$, whereas $d_{G'}(v, w) = d_G(v, w) - (b-a) + 1 < d_{G'}(v, w')$. In consequence, there exists a vertex $w''$ in a shortest $v_k - w'$ path such that $d_{G'}(v, w) = d_{G'}(v, w'')$, so $w$ is not 1- resolvable by $\{v\}$ in $G'$.

Summing up the cases above, we have that every $w \in \{v_1, \ldots, v_n\}$ is not 1-resolvable by $\{v\}$ in $G'$. □

## B Proof of Theorem 12

**Theorem 12.** *Let $G = (V, E)$ be a simple and connected graph, $\{v\}$ a 1-antiresolving set of $G$, and $v_1 \dots v_m$ an eccentricity path of $v$ where $v_1 = v$. Let $i$ and $j$ be the lowest and largest positive integers, respectively, such that $v_i$ and $v_j$ are 1-resolvable by $\{v\}$ in $G$. Let $a, b \in \{1, \dots, m\}$ and let $G' = (V, E \cup \{(v_a, v_b)\})$ be the graph resulting from the addition of the edge $(v_a, v_b)$. Every vertex $v_i$ with $i \in \{1, \dots, m\}$ is not 1-resolvable by $\{v\}$ in $G'$ if the following conditions hold:*

*i.* $1 \leq a \leq i - 1$.

*ii.* $a + 2 \leq b \leq m$.

*iii.* *If* $b - a = 2r + 1$, *then* $j - b \leq r \leq m - b$.

*Proof.* We have that $b - a = 2r + 1$ and $j - b \leq r \leq m - b$. We will rely on the fact that the subgraph of $G'$ induced by the set $\{v_a, \dots, v_{b+r}\}$ is a $(2r + 2, r)$-tadpole, to prove that a vertex $w \in \{v_1, \dots, v_m\}$ is not 1-resolvable by $\{v\}$ in $G'$. To that end, we will analyse the following three scenarios regarding the position of the vertex $w$ in the eccentricity path $v_1 \dots v_m$ (see Figure 10):

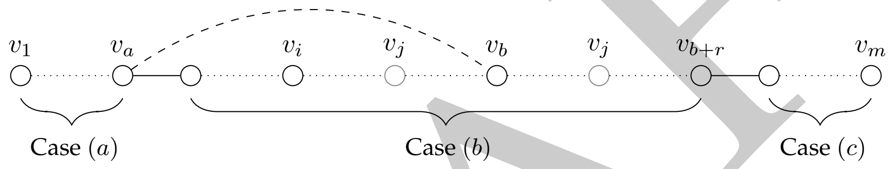

Figure 10: An eccentricity path $v_1 \dots v_m$ within the graph $G$. The dashed line represents the edge added to obtain the graph $G'$. Note that two possible positions for the vertex $v_j$ are depicted.

(a) $w \in \{v_1, \dots, v_a\}$. This case is equivalent to Cases 1(a) and 2(a) in the proof of Theorem 11. Thus, by analogy, we have that $w$ is not 1-resolvable by $\{v\}$ in $G'$.

(b) $w \in \{v_{a+1}, \dots, v_{b+r}\}$. First, note that all vertices in the eccentricity path $v_1 \dots, v_m$ that are 1-resolvable by $\{v\}$ in $G$ fall into this case, as $a + 1 \leq i$ and $j \leq b + r \leq m$. Let $H$ be the subgraph of $G'$ induced by $\{v_a, v_{a+1}, \dots, v_{b+r}\}$. As we pointed out above, $H$ is a $(2r + 2, r)$-tadpole. Since $v_b$ is the sole degree-3 vertex of $H$, we have that $\{v_a\}$ is a 2-antiresolving set of $H$ by Theorem 9. In consequence, there exists a vertex $w' \in \{v_{a+1}, \dots, v_{b+r}\} \setminus \{w\}$ such that $d_H(v_a, w) = d_H(v_a, w')$. Since $d_{G'}(v, x) = d_{G'}(v, v_a) + d_{G'}(v_a, x) = a - 1 + d_H(v_a, x)$ for every $x \in V(H)$, we can conclude that $d_{G'}(v, w) = a - 1 + d_H(v_a, w) = a - 1 + d_H(v_a, w') = d_{G'}(v, w')$, so $w$ is not 1-resolvable by $\{v\}$ in $G'$.

(c) $w \in \{v_{b+r+1}, \dots, v_m\}$ (this case occurs only if $b + r < m$). If $b \leq j$, this case is equivalent to Case 1(c) in the proof of Theorem 11, otherwise it is equivalent to Case 2(c). In either situation, we have that, by analogy, $w$ is not 1-resolvable by $\{v\}$ in $G'$.

Summing up the cases above, we have that every $w \in \{v_1, \dots, v_m\}$ is not 1-resolvable by $\{v\}$ in $G'$. □