Noname manuscript No.
(will be inserted by the editor)

Robust active attacks on social graphs

Sjouke Mauw · Yunior Ramírez-Cruz · Rolando Trujillo-Rasua

Abstract In order to prevent the disclosure of privacy-sensitive data, such as names and relations between users, social network graphs have to be anonymised before publication. Naive anonymisation of social network graphs often consists in deleting all identifying information of the users, while maintaining the original graph structure. Various types of attacks on naively anonymised graphs have been developed. Active attacks form a special type of such privacy attacks, in which the adversary enrols a number of fake users, often called *sybils*, to the social network, allowing the adversary to create unique structural patterns later used to re-identify the sybil nodes and other users after anonymisation. Several studies have shown that adding a small amount of noise to the published graph already suffices to mitigate such active attacks. Consequently, active attacks have been dubbed a negligible threat to privacy-preserving social graph publication. In this paper, we argue that these studies unveil shortcomings of specific attacks, rather than inherent problems of active attacks as a general strategy. In order to support this claim, we develop the notion of a *robust active attack*, which is an active attack that is resilient to small perturbations of the social network graph. We formulate the design of robust active attacks as an optimisation problem and we give definitions of robustness for different stages of the active attack strategy. Moreover, we introduce various heuristics to achieve these notions of robustness and experimentally show that the new robust attacks are considerably more resilient than the original ones, while remaining at the same level of feasibility.

Keywords: privacy-preserving social graph publication, active attacks

1 Introduction

Data is useful. Science heavily relies on data to (in)validate hypotheses, discover new trends, tune up mathematical and computational models, etc. In other words, data collection and analysis is helping to cure diseases, build more efficient and environmentally-friendly buildings, take socially-responsible decisions, understand our needs and those of the planet where we live. Despite these indisputable benefits, it is also a fact that data contains personal and potentially sensitive information, and this is where privacy and usefulness should be considered as a whole.

A massive source of personal information is currently being handled by online social networks. People's life is often transparently reflected on popular social network platforms, such as Facebook, Twitter and Youtube. Therefore, releasing social network data for further study comes with a commitment to ensure that users remain anonymous. Anonymity, however, is remarkably hard to

S. Mauw, Y. Ramírez-Cruz
CSC, SnT, University of Luxembourg

R. Trujillo-Rasua
School of Information Technology, Deakin University, Geelong, Australia
Centre for Cyber Security Research and Innovation

achieve. Even a simple social graph, where an account consists of a user's pseudonym only and its relation to other accounts, allows users to be *re-identified* by just considering the number of relations they have (Liu and Terzi, 2008).

The use of pseudonyms is insufficient to guarantee anonymity. An attacker can cross-reference information from other sources, such as the number of connections, to find out the real user behind a pseudonym. Taking into account the type of information an attacker may have, called *background* or *prior* knowledge, is thus a common practice in anonymisation models. In a social graph, the adversary's background knowledge is regarded as any subgraph that is isomorphic to a subgraph in the original social graph. Various works bound the adversary's background knowledge to a specific family of graphs. For example, the adversary model introduced by Liu and Terzi relies on knowing the degrees of the victim vertices, thus in this case the background knowledge is fully defined by star graphs[1]. Others assume that an adversary may know the subgraphs induced by the neighbours of their victims (Zhou and Pei, 2008), an extended vicinity (Zou et al., 2009), and so on.

A rather different notion of background knowledge was introduced by Backstrom et al. (2007). They describe an adversary able to register several (fake) accounts to the network, called *sybil accounts*. The sybil accounts establish links between themselves and also with the victims. Therefore, in Backstrom et al.'s attack to a social graph $G = (V, E)$, the adversary's background knowledge is the induced subgraph formed by the sybil accounts in G joined with the connections to the victims.

The adversary introduced by Backstrom et al. is said to be *active*, because he influences the structure of the social network. Previous authors have claimed that active attacks are either unfeasible or detectable. Such a claim is based on two observations. First, inserting many sybil nodes is hard, and they may be detected and removed by sybil detection techniques (Narayanan and Shmatikov, 2009). Second, active attacks have been reported to suffer from low resilience, in the sense that the attacker's ability to successfully recover the sybil subgraph and re-identify the victims is easily lost after a relatively small number of (even arbitrary) changes are introduced in the network (Ji et al., 2015; Mauw et al., 2016, 2018a,b). As a consequence, active attacks have been largely overlooked in literature. Backstrom et al. argue for the feasibility of active attacks, showing that proportionally few sybil nodes (in the order of $\log_2 n$ nodes for networks of order n) are sufficient for compromising any legitimate node. This feature of active attacks is relevant in view of the fact that sybil defence mechanisms do not attempt to remove every sybil node, but to limit their number to no more than $\log_2 n$ (Yu et al., 2006, 2008), which entails that sufficiently capable sybil subgraphs are likely to go unnoticed by sybil defences. The second claim, that of lack of resilience to noisy releases, is the main focus of this work.

Contributions. In this paper we show that active attacks do constitute a serious threat for privacy-preserving publication of social graphs. We do so by proposing the first active attack strategy that features two key properties. Firstly, it can effectively re-identify users with a small number of sybil accounts. Secondly, it is resilient, in the sense that it resists not only the introduction of reasonable amounts of noise in the network, but also the application of anonymisation algorithms specifically designed to counteract active attacks. The new attack strategy is based on new notions of robustness for the sybil subgraph and the set of fingerprints, as well as noise-tolerant algorithms for sybil subgraph retrieval and re-identification. The comparison of the robust active attack strategy to the original active attack is facilitated by the introduction of a novel framework of study, which views an active attack as an attacker-defender game.

The remainder of this paper is structured as follows. Section 2 examines the literature on social network privacy with a clear focus on active attacks. As part of the problem formulation, we enunciate our adversarial model in the form of an attacker-defender game in Section 3. Then, the new notions of robustness are introduced in Section 4, and their implementation is discussed in Section 5. Finally, we experimentally evaluate our proposal in Section 6 and give our conclusions in Section 7.

[1] A star graph is a tree of depth one.

2 Related work

Privacy attacks on social networks exploit structural knowledge about the victims for re-identifying them in a released version of the social graph. These attacks can be divided in two categories, according to the manner in which the adversary obtains the knowledge used to re-identify the victims. On the one hand, *passive attacks* rely on existing knowledge, which can be collected from publicly available sources, such as the public view of another social network where the victims are known to have accounts. The use of this type of information was demonstrated by Narayanan and Shmatikov (2009), who used information from Flickr to re-identify users in a pseudonymised Twitter graph.

On the other hand, *active attacks* rely on the ability to alter the structure of the social graph, in such a way that the unique structural properties allowing to re-identify the victims after publication are guaranteed to hold, and to be known by the adversary. As we discussed previously, the active attack methodology was introduced by Backstrom et al. (2007). They proposed to use sybil nodes to create re-identifiable patterns for the victims, in the form of fingerprints defined by sybil-to-victim edges. Under this strategy, they proposed two attacks, the walk-based attack and the cut-based attack. The difference between both attacks lies in the structure given to the sybil subgraph for facilitating its retrieval after publication. In the walk-based attack, a long path linking all the sybil nodes in a predefined order is created, with remaining inter-sybil edges randomly generated. In the cut-based attack, a subset of the sybil nodes are guaranteed to be the only cut vertices linking the sybil subgraph and the rest of the graph. Interestingly, Backstrom et al. also study a passive version of these attacks, where fingerprints are used as identifying information, but no sybil nodes are inserted. Instead, they model the situation where legitimate users turn rogue and collude to share their neighbourhood information in order to retrieve their own weakly induced subgraph and re-identify some of their remaining neighbours. However, the final conclusion of this study is that the active attack is more capable because sybil nodes can better guarantee to create a uniquely retrievable subgraph and unique fingerprints.

A hybrid attack strategy was proposed by Peng et al. (2012, 2014). This attack is composed of two stages. First, a small-scale active attack is used to re-identify an initial set of victims, and then a passive attack is used to iteratively enlarge the set of re-identified victims with neighbours of previously re-identified victims. Because of the order in which the active and the passive phases are executed, the success of the initial active attack is critical to the entire attack. Beyond that interplay between active and passive attacks, Peng et al. do not introduce improvements over the original active attack strategy.

There exist a large number of anonymisation methods for the publication of social graphs that can resist privacy attacks, as those described previously. They can be divided into three categories: those that produce a perturbed version of the original graph (Liu and Terzi, 2008; Zhou and Pei, 2008; Zou et al., 2009; Cheng et al., 2010; Lu et al., 2012; Casas-Roma et al., 2013; Chester et al., 2013; Wang et al., 2014; Ma et al., 2015; Salas and Torra, 2015; Rousseau et al., 2017; Casas-Roma et al., 2017), those that generate a new synthetic graph sharing some statistical properties with the original graph (Hay et al., 2008; Mittal et al., 2013; Liu and Mittal, 2016; Jorgensen et al., 2016), and those that output some aggregate statistic of the graph without releasing the graph itself, e.g. differentially private degree correlation statistics (Sala et al., 2011), degree distributions (Karwa and Slavković, 2012), subgraph counts (Zhang et al., 2015), etc. Active attacks, both the original formulation and the robust version presented in this paper, are relevant to the first type of releases. In this context, a number of methods have been proposed aiming to transform the graph into a new one satisfying some anonymity property based on the notion of k-anonymity (Samarati, 2001; Sweeney, 2002). Examples of this type of anonymity properties for passive attacks are k-degree anonymity (Liu and Terzi, 2008), k-neighbourhood anonymity (Zhou and Pei, 2008) and k-automorphism (Zou et al., 2009). For the case of active attacks, the notion of (k, ℓ)-anonymity was introduced by Trujillo-Rasua and Yero (2016). A (k, ℓ)-anonymous graph guarantees that an active attacker with the ability to insert up to ℓ sybil nodes in the network will still be unable to distinguish any user from at least other $k-1$ users, in terms of their distances to the sybil nodes. Several relaxations of the notion of (k, ℓ)-anonymity were introduced

by Mauw et al. (2018b). The notion of (k, ℓ)-adjacency anonymity accounts for the unlikelihood of the adversary to know all distances in the original graph, whereas $(k, \Gamma_{G,\ell})$-anonymity models the protection of the victims only from vertex subsets with a sufficiently high re-identification probability and $(k, \Gamma_{G,\ell})$-adjacency anonymity combines both criteria.

Anonymisation methods based on the notions of (k, ℓ)-anonymity, $(k, \Gamma_{G,\ell})$-anonymity and $(k, \Gamma_{G,\ell})$-adjacency anonymity were introduced by Mauw et al. (2016, 2018a,b). As we discussed above, despite the fact that these methods only give a theoretical privacy guarantee against adversaries with the capability of introducing a small number of sybil nodes, empirical results show that they are in fact capable of thwarting attacks leveraging larger numbers of sybil nodes. These results are in line with the observation that random perturbations also thwart active attacks in their original formulation (Narayanan and Shmatikov, 2009; Ji et al., 2015). In contrast, our robust active attack strategy performs significantly better in the presence of random perturbation, as we demonstrate in Section 6.

In the context of obfuscation methods, which aim to publish a new version of the social graph with randomly added perturbations, Xue et al. (2012) assess the possibility of the attacker leveraging the knowledge about the noise generation to launch what they call a *probabilistic* attack. In their work, Xue et al. provided accurate estimators for several graph parameters in the noisy graphs, to support the claim that useful computations can be conducted on the graphs after adding noise. Among these estimators, they included one for the degree sequence of the graph. Then, noting that an active attacker can indeed profit from this estimator to strengthen the walk-based attack, they show that after increasing the perturbation by a sufficiently small amount this attack also fails. Although the probabilistic attack presented by Xue et al. features some limited level of noise resilience, it is not usable as a general strategy, because it requires the noise to follow a specific distribution and the parameters of this distribution to be known by the adversary. Our definition of robust attack makes no assumptions about the type of perturbation applied to the graph. It is also worth noting, in the context of noise addition methods, that anonymisation algorithms based on privacy properties for passive attacks, such as k-degree anonymity or k-neighbourhood anonymity, can in some cases thwart an active attack. This may happen if such a method introduces a sufficiently large amount of changes in the graph. However, these anonymity notions do not offer formal privacy guarantees against active attacks, because they target adversary models based on different forms of background knowledge. In other words, if some of these algorithms happen to thwart an active attack, it will be a side effect of the noise that it introduced rather than a consequence of the privacy property imposed on the graph.

Finally, we point out that the active attack strategy shares some similarities with graph watermarking methods (Collberg et al., 2003; Zhao et al., 2015; Eppstein et al., 2016). The purpose of graph watermarking is to release a graph containing embedded instances of a small subgraph, the *watermark*, that can be easily retrieved by the graph publisher, while remaining imperceptible to others and being hard to remove or distort. Note that the goals of the graph owner and the adversary are to some extent inverted in graph watermarking, with respect to active attacks. Moreover, since the graph owner knows the entire graph, he can profit from this knowledge for building the watermark. However, during the sybil subgraph creation phase of an active attack, only a partial view of the social graph is available to the attacker. The next section will make it easier to understand the exact limitations and capabilities of the active adversary, as well as those of the defender.

3 Adversarial model

We design a game between an attacker $\mathcal{A}$ and a defender $\mathcal{D}$. The goal of the attacker is to identify the victim nodes after pseudonymisation and transformation of the graph by the defender. We first introduce the necessary graph theoretical notation, and then formulate the three stages of the attacker-defender game.

3.1 Notation and terminology

We use the following standard notation and terminology. Additional notation that may be needed in other sections of the paper will be introduced as needed.

- A *graph* G is represented as a pair (V, E), where V is a set of vertices (also called nodes) and $E \subseteq V \times V$ is a set of edges. The vertices of G are denoted by V_G and its edges by E_G. As we will only consider undirected graphs, we will consider an edge (v, w) as an unordered pair. We will use the notation $\mathcal{G}$ for the set of all graphs.
- An *isomorphism* between two graphs $G = (V, E)$ and $G' = (V', E')$ is a bijective function $\varphi\colon V \to V'$, such that $\forall v_1, v_2 \in V\colon (v_1, v_2) \in E \iff (\varphi(v_1), \varphi(v_2)) \in E'$. Two graphs are *isomorphic*, denoted by $G \simeq_\varphi G'$, or briefly $G \simeq G'$, if there exists an isomorphism φ between them. Given a subset of vertices $S \subseteq V$, we will often use φS to denote the set $\{\varphi(v) | v \in S\}$.
- The set of *neighbours* of a set of nodes $W \subseteq V$ is defined by $N_G(W) = \{v \in V \setminus W \mid \exists w \in W\colon (v, w) \in E \vee (w, v) \in E\}$. If $W = \{w\}$ is a singleton set, we will write $N_G(w)$ for $N_G(\{w\})$. The *degree* of a vertex $v \in V$, denoted as $\delta_G(v)$, is defined as $\delta_G(v) = |N_G(v)|$.
- Let $G = (V, E)$ be a graph and let $S \subseteq V$. The *weakly-induced subgraph* of S in G, denoted by $\langle S \rangle^w_G$, is the subgraph of G with vertices $S \cup N_G(S)$ and edges $\{(v, v') \in E \mid v \in S \vee v' \in S\}$.

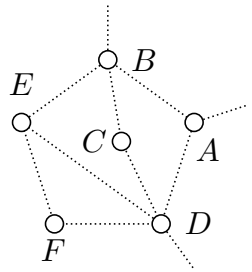
3.2 The attacker-defender game

The attacker-defender game starts with a graph $G = (V, E)$ representing a snapshot of a social network. The attacker knows a subset of the users, but not the connections between them. This is a common scenario in online social networks such as Facebook, where every user has the choice of not showing her friend list, even to her friends or potential adversaries who, in principle, do know that the victim is enrolled in the network. In other types of social networks, e.g. in e-mail networks such as Gmail or messaging networks such as WhatsApp, the existence of relations (determined in this case by the action of exchanging messages) is by default not known, even if the adversary knows the victim's e-mail address or phone number. Figure 1(a) exemplifies the initial state of a small network, where capital letters represent the real identities of the users and dotted lines represent the relations existing between them, which are not known to the adversary.

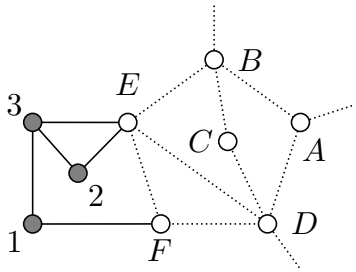
Before a pseudonymised graph is released, the attacker manages to enrol sybil accounts in the network and establish links with the victims, as depicted in Figure 1(b), where sybil accounts are represented by dark-coloured nodes and the edges known to the adversary (because they were created by her) are represented by solid lines. The goal of the attacker is to later re-identify the victims in order to learn information about them. Coming back to the real-life scenarios discussed before, creating accounts in Facebook or Gmail is trivial, and social engineering may be used to get the victims to accept a friend request or answer an e-mail.

When the defender decides to publish the graph, she anonymises it by removing the real user identities, or replacing them with pseudonyms, and possibly perturbing the graph. In Figure 1(c) we illustrate the pseudonymisation process of the graph in Figure 1(b). The pseudonymised graph contains information that the attacker wishes to know, such as the existence of relations between users, but the adversary cannot directly learn this information, as the identities of all the vertices are hidden, including those of the sybil nodes themselves. Thus, after the pseudonymised graph is published, the attacker analyses the graph to first re-identify her own sybil accounts, and then the victims (see Figure 1(d)). This allows her to acquire new information, which was supposed to remain private, such as the fact that E and F are friends on Facebook, or e-mail each other via Gmail. Note that by "publishing an anonymised graph", we do not necessarily mean publishing the entire graph underlying large networks such as Facebook or Gmail. This is unlikely to occur in practice. However, sanitised samples from graphs of this type, covering users in a particular group, such as students of a particular school, have indeed been published in the past for research purposes (Guimera et al., 2003; Panzarasa et al., 2009).

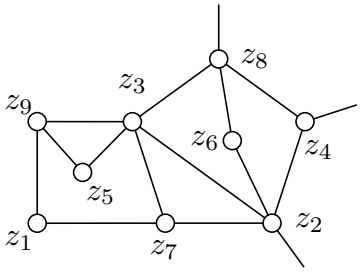
In what follows, we formalise the three stages of the attacker-defender game, assuming an initial graph $G = (V, E)$.



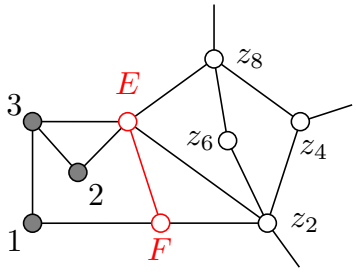
(a) A fragment of the original graph.



(b) The sybil nodes are added and fingerprints are created for the victims.



(c) Pseudonymised graph after release.



(d) The attacker subgraph is recovered and the victims are re-identified.

Fig. 1 The attacker-defender game.

1. *Attacker subgraph creation.* The attacker constructs a set of *sybil* nodes $S = \{x_1, x_2, \ldots, x_{|S|}\}$, such that $S \cap V = \emptyset$ and a set of edges $F \subseteq (S \times S) \cup (S \times V) \cup (V \times S)$. It clearly follows that $E \cap F = \emptyset$. We call $G^+ = (V \cup S, E \cup F)$ the *sybil-extended* graph of G. The attacker does not know the complete graph G^+, but he knows $\langle S \rangle^w_{G^+}$, the weakly-induced subgraph of S in G^+. We say that $\langle S \rangle^w_{G^+}$ is the *attacker subgraph*. The attacker subgraph creation has two substages:
   (a) *Creation of inter-sybil connections.* A unique (with high probability) and efficiently retrievable connection pattern is created between sybil nodes to facilitate the attacker's task of retrieving the sybil subgraph at the final stage.
   (b) *Fingerprint creation.* For a given *victim* vertex $y \in N_{G^+}(S) \setminus S$, we call the victim's neighbours in S, i.e $N_{G^+}(y) \cap S$, its *fingerprint*. Considering the set of victim vertices $Y = \{y_1, \ldots, y_m\}$, the attacker ensures that $N_{G^+}(y_i) \cap S \neq N_{G^+}(y_j) \cap S$ for every $y_i, y_j \in Y$, $i \neq j$.
2. *Anonymisation.* The defender obtains G^+ and constructs an isomorphism φ from G^+ to φG^+. We call φG^+ the *pseudonymised* graph. The purpose of pseudonymisation is to remove all personally identifiable information from the vertices of G. Next, given a non-deterministic procedure t that maps graphs to graphs, known by both $\mathcal{A}$ and $\mathcal{D}$, the defender applies transformation t to φG^+, resulting in the *transformed* graph $t(\varphi G^+)$. The procedure t modifies φG^+ by adding and/or removing vertices and/or edges.
3. *Re-identification.* After obtaining $t(\varphi G^+)$, the attacker executes the re-identification attack in two stages.
   (a) *Attacker subgraph retrieval.* Determine the isomorphism φ restricted to the domain of sybil nodes S.
   (b) *Fingerprint matching.* Determine the isomorphism φ restricted to the domain of victim nodes $\{y_1, y_2, \ldots, y_m\}$.

As established by the last step of the attacker-defender game, we consider the adversary to succeed if she effectively determines the isomorphism φ restricted to the domain of victim nodes $\{y_1, y_2, \ldots, y_m\}$. That is, when the adversary re-identifies all victims in the anonymised graph.

4 Robust active attacks

This section formalises robust active attacks. We provide mathematical formulations, in the form of optimisation problems, of the attacker's goals in the first and third stages. In particular, we address three of the subtasks that need to be accomplished in these stages: fingerprint creation, attacker subgraph retrieval and fingerprint matching.

4.1 Robust fingerprint creation

Active attacks, in their original formulation (Backstrom et al., 2007), aimed at re-identifying victims in pseudonymised graphs. Consequently, the uniqueness of every fingerprint was sufficient to guarantee success with high probability, provided that the attacker subgraph is correctly retrieved. Moreover, several types of randomly generated attacker subgraphs can indeed be correctly and efficiently retrieved, with high probability, after pseudonymisation. The low resilience reported for this approach when the pseudonymised graph is perturbed by applying an anonymisation method (Mauw et al., 2018a,b, 2016) or by introducing arbitrary changes (Ji et al., 2015), comes from the fact that it relies on finding exact matches between the fingerprints created by the attacker at the first stage of the attack and their images in $t(\varphi G^+)$. The attacker's ability to find such exact matches is lost even after a relatively small number of perturbations is introduced by t.

Our observation is that setting for the attacker the goal of obtaining the exact same fingerprints in the perturbed graph is not only too strong, but more importantly, not necessary. Instead, we argue that it is sufficient for the attacker to obtain a set of fingerprints that is close enough to the original set of fingerprints, for some notion of closeness. Given that a fingerprint is a set of vertices, we propose to use the cardinality of the symmetric difference of two sets to measure the distance between fingerprints. The symmetric difference between two sets X and Y, denoted by $X \triangledown Y$, is the set of elements in $X \cup Y$ that are not in $X \cap Y$. We use $d(X, Y)$ to denote $|X \triangledown Y|$.

Our goal at this stage of the attack is to create a set of fingerprints satisfying the following property.

Definition 1 (Robust set of fingerprints) Given a set of victims $\{y_1, \ldots, y_m\}$ and a set of sybil nodes S in a graph G^+, the set of fingerprints $\{F_1, \ldots, F_m\}$ with $F_i = N_{G^+}(y_i) \cap S$ is said to be *robust* if it maximises

$$\min_{1 \leq i < j \leq m} \{d(F_i, F_j)\}. \tag{1}$$

The property above ensures that the lower bound on the distance between any pair of fingerprints is maximal. In what follows, we will refer to the lower bound defined by Equation (1) as *minimum separation* of a set of fingerprints. For example, in Figure 1(b), the fingerprint of the vertex E with respect to the set of attacker vertices $\{1, 2, 3\}$ is $\{2, 3\}$, and the fingerprint of the vertex F is $\{1\}$. This gives a minimum separation between the two victim's fingerprints equal to $|\{2, 3\} \triangledown \{1\}| = |\{1, 2, 3\}| = 3$, which is maximum. Therefore, given attacker vertices $\{1, 2, 3\}$, the set of fingerprints $\{\{2, 3\}, \{1\}\}$ is robust for the set of victim nodes $\{E, F\}$.

Next we prove that, if the distance between each original fingerprint F and the corresponding anonymised fingerprint φF is less than half the minimum separation, then the distance between F and any other anonymised fingerprint, say $\varphi F'$, is strictly larger than half the minimum separation.

Theorem 1 *Let S be the set of sybil nodes, let $\{y_1, \ldots, y_m\}$ be the set of victims and let $\{F_1, \ldots, F_m\}$ be their fingerprints with minimum separation δ. Let F'_i be the fingerprint of φy_i in the anonymised graph $t(\varphi G^+)$, for $i \in \{1, \ldots, m\}$. Then,*

$$\forall i \in \{1,\ldots,m\}\colon d(\varphi F_i, F_i') < \delta/2 \implies \forall i,j \in \{1,\ldots,m\}\colon i \neq j \implies d(\varphi F_i, F_j') > \delta/2,$$

Proof In order to achieve a contradiction, we assume that $d(\varphi F_i, F_j') \leq \delta/2$ for some $i,j \in \{1,\ldots,m\}$ with $i \neq j$. Because $d(\varphi F_j, F_j') < \delta/2$, we have $d(\varphi F_i, F_j') + d(\varphi F_j, F_j') < \delta$. By the triangle inequality we obtain that $d(\varphi F_i, \varphi F_j) \leq d(\varphi F_i, F_j') + d(\varphi F_j, F_j') < \delta$. Hence $d(\varphi F_i, \varphi F_j)$ is lower than the minimum separation of $\{F_1,\ldots,F_m\}$, which yields a contradiction given that $d(\varphi F_i, \varphi F_j) = d(F_i, F_j) \geq \delta$. □

We exploit Theorem 1 later in the fingerprint matching step through the following corollary. If $\delta/2$ is the maximum distance shift from an original fingerprint F_i of y_i to the fingerprint F_i' of y_i in the perturbed graph, then for every $F \in \{F_1',\ldots,F_m'\}$ it holds that $d(F, \varphi F_i) < \delta/2 \iff F = \varphi F_i$. In other words, given a set of victims for which a set of fingerprints needs to be defined, the larger the minimum separation of these fingerprints, the larger the number of perturbations that can be tolerated in $t(\varphi G^+)$, while still being able to match the perturbed fingerprints to their correct counterparts in G^+.

As illustrated earlier in our running example, the fingerprints of E and F are $\{2,3\}$ and $\{1\}$, respectively, which gives a minimum separation of $\delta = 3$. Theorem 1 states that, if after anonymisation of the graph, the fingerprints of E and F become, say $\{2\}$ and $\{1,2\}$, respectively, then it must hold that $|\{2,3\} \triangledown \{1,2\}| > 3/2$ and $|\{1\} \triangledown \{2\}| > 3/2$, while $|\{2,3\} \triangledown \{2\}| < 3/2$ and $|\{1\} \triangledown \{1,2\}| < 3/2$. This makes it easy to match the original fingerprint, say $\{2,3\}$, with the correct perturbed fingerprint $\{2\}$ by calculating their distance and verifying that it remains below the threshold $\delta/2$. In Subsection 5.1, we will describe an efficient algorithm for addressing this optimisation problem.

4.2 Robust attacker subgraph retrieval

Let $\mathcal{C} = \{\langle X \rangle^w_{t(\varphi G^+)} \mid X \subseteq V_{t(\varphi G^+)}, |X| = |S|, \langle X \rangle^w_{t(\varphi G^+)} \cong \langle S \rangle^w_{G^+}\}$ be the set of all subgraphs isomorphic to the attacker subgraph $\langle S \rangle^w_{G^+}$ and weakly induced in $t(\varphi G^+)$ by a vertex subset of cardinality $|S|$. The original active attack formulation assumes that $|\mathcal{C}| = 1$ and that the subgraph in $\mathcal{C}$ is the image of the attacker subgraph after pseudonymisation. This assumption, for example, holds on the pseudonymised graph in Figure 1(c), but it rarely holds on perturbed graphs. In fact, $\mathcal{C}$ becomes empty by simply adding an edge between any pair of attacker nodes, which makes the attack fail quickly when increasing the amount of perturbation.

To account for the occurrence of perturbations in releasing $t(\varphi G^+)$, we introduce the notion of robust attacker subgraph retrieval. Rather than limiting the retrieval process to finding an exact match of the original attacker subgraph, we consider that it is enough to find a sufficiently similar subgraph, thus adding some level of noise-tolerance. By "sufficiently similar", we mean a graph that minimises some graph dissimilarity measure $\Delta\colon \mathcal{G} \times \mathcal{G} \to \mathbb{R}^+$ with respect to $\langle S \rangle^w_{G^+}$. The problem is formulated as follows.

Definition 2 (Robust attacker subgraph retrieval problem) Given a graph dissimilarity measure $\Delta\colon \mathcal{G} \times \mathcal{G} \to \mathbb{R}^+$, and a set S of sybil nodes in the graph G^+, find a set $S' \subseteq V_{t(\varphi G^+)}$ that minimises

$$\Delta(\langle S' \rangle^w_{t(\varphi G^+)}, \langle S \rangle^w_{G^+}). \tag{2}$$

A number of graph (dis)similarity measures have been proposed in the literature (Sanfeliu and Fu, 1983; Bunke, 2000; Backstrom et al., 2007; Fober et al., 2013; Mallek et al., 2015). Commonly, the choice of a particular measure is *ad hoc*, and depends on the characteristics of the graphs being compared. In Subsection 5.2, we will describe a measure that is efficiently computable and exploits the known structure of $\langle S \rangle^w_{G^+}$, by separately accounting for inter-sybil and sybil-to-non-sybil edges. Along with this dissimilarity measure, we provide an algorithm for constructively finding a solution to the problem enunciated in Definition 2.

4.3 Robust fingerprint matching

As established by the attacker-defender game discussed in Section 3, fingerprint matching is the last stage of the active attack. Because it clearly relies on the success of the previous steps, we make the following two assumptions upfront.

1. We assume that the robust sybil subgraph retrieval procedure succeeds, i.e. that $\varphi S = S'$ where S' is the set of sybil nodes obtained in the previous step.
2. Given the original set of victims Y, we assume that the set of vertices in the neighbourhood of S' contains those in φY, i.e. $\varphi Y \subseteq N_{t(\varphi G^+)}(S') \setminus S'$, otherwise S' is insufficient information to achieve the goal of re-identifying all victim vertices.

Given the correct set of sybil nodes S' and a set of potential victims $Y' = \{y'_1, \ldots, y'_n\} = N_{t(\varphi G^+)}(S') \setminus S'$, the re-identification process consists in determining the isomorphism φ restricted to the vertices in Y'. Next we define re-identification as an optimisation problem, and after that we provide sufficient conditions under which a solution leads to correct identification.

Definition 3 (Robust re-identification problem) Let S and S' be the set of sybil nodes in the original and anonymised graph, respectively. Let $\{y_1, \ldots, y_m\}$ be the victims in G^+ with fingerprints $F_1 = N_{G^+}(y_1) \cap S, \ldots, F_m = N_{G^+}(y_m) \cap S$. The *robust re-identification problem* consists in finding an isomorphism $\phi\colon S \to S'$ and subset $\{z_1, \ldots, z_m\} \subseteq N_{t(\varphi G^+)}(S') \setminus S'$ that minimises

$$\|(d(\phi F_1, N_{t(\varphi G^+)}(z_1) \cap S'), \ldots, d(\phi F_m, N_{t(\varphi G^+)}(z_m) \cap S'))\|_\infty. \tag{3}$$

where $\|.\|_\infty$ stands for the infinity norm.

Optimising the infinity norm gives the lowest upper bound on the distance between an original fingerprint and the fingerprint of a vertex in the perturbed graph. This is useful towards the goal of correctly re-identifying all victims. However, should the adversary aims at re-identifying at least one victim with high probability, then other plausible objective functions can be used, such as the Euclidean norm.

As stated earlier, our intention is to exploit the result of Theorem 1, provided that the distance between original and perturbed fingerprints is lower than $\delta/2$, where δ is the minimum separation of the original set of fingerprints. This is one out of three conditions that we prove sufficient to infer a correct mapping φ from a solution to the robust re-identification problem, as stated in the following result.

Theorem 2 *Let $\phi\colon S \to S'$ and $\{z_1, \ldots, z_m\}$ be a solution to the robust re-identification problem defined by the set of sybil nodes S in the original graph G^+, the set of sybil nodes S' in the anonymised graph $t(\varphi G^+)$, and the set of victims $\{y_1, \ldots, y_m\}$ in G^+. Let $\{F_1, \ldots, F_m\}$ be the set of fingerprints of $\{y_1, \ldots, y_m\}$ and δ its minimum separation. If the following three conditions hold:*

1. $\forall x \in S\colon \phi(x) = \varphi(x)$
2. $\{\varphi(y_1), \ldots, \varphi(y_m)\} = N_{t(\varphi G^+)}(S') \setminus S'$
3. *For every $y_i \in \{y_1, \ldots, y_m\}$, $d(\varphi F_i, F'_i) < \delta/2$ where $F'_i = N_{t(\varphi G^+)}(\varphi(y_i)) \cap S'$,*

then $\varphi(y_i) = z_i$ for every $i \in \{1, \ldots, m\}$.

Proof From the third condition we obtain that the correct mapping φ satisfies

$$\max\{d(\varphi F_1, F'_1), \ldots, d(\varphi F_m, F'_m)\} < \delta/2.$$

Now, the second condition gives that $\{\varphi(y_1), \ldots, \varphi(y_m)\} = \{z_1, \ldots, z_m\}$. This means that, for every $i \in \{1, \ldots, m\}$, $F'_i = N_{t(\varphi G^+)}(z_j) \cap S'$ for some $j \in \{1, \ldots, m\}$. Let f be an automorphism in $\{1, \ldots, m\}$ such that $F'_i = N_{t(\varphi G^+)}(z_{f(i)}) \cap S'$ for every $i \in \{1, \ldots, m\}$. We use f^{-1} to denote the inverse of f. Then, considering that $\phi F_i = \varphi F_i$ for every $i \in \{1, \ldots, m\}$ (first condition), we obtain the following equalities.

$$\max\{d(\phi F_1, N_{t(\varphi G^+)}(z_1) \cap S'), \ldots, d(\phi F_m, N_{t(\varphi G^+)}(z_m) \cap S')\} =$$
$$\max\{d(\varphi F_1, N_{t(\varphi G^+)}(z_1) \cap S'), \ldots, d(\varphi F_m, N_{t(\varphi G^+)}(z_m) \cap S')\} =$$
$$\max\{d(\varphi F_1, F'_{f^{-1}(1)}), \ldots, d(\varphi F_m, F'_{f^{-1}(m)})\}$$

Considering Theorem 1, we obtain that for every $i, j \in \{1, \ldots, m\}$ with $i \neq j$ it holds that $d(\varphi F_i, F'_j) > \delta/2$. Therefore, if f is not the trivial automorphism, i.e. $f(i) = i \; \forall i \in \{1, \ldots, m\}$, then $\max\{d(\varphi F_1, F'_{f^{-1}(1)}), \ldots, d(\varphi F_m, F'_{f^{-1}(m)})\} > \delta/2$. This implies that,

$$\max\{d(\phi F_1, N_{t(\varphi G^+)}(z_1) \cap S'), \ldots, d(\phi F_m, N_{t(\varphi G^+)}(z_m) \cap S')\} > \delta/2,$$

However, this contradicts the optimality of the solution ϕ and $\{z_1, \ldots, z_m\}$. Therefore, f must be the trivial automorphism, which concludes the proof. □

In Theorem 2, the first condition states that the adversary succeeded on correctly identifying each of her own sybil nodes in the perturbed graph. That is to say, the adversary retrieved the mapping φ restricted to the set of victims. This is clearly an important milestone in the attack as victim's fingerprints are based on such mapping. The second condition says that the neighbours of the sybil vertices remained the same after perturbation. As a result, the adversary knows that $\{z_1, \ldots, z_m\}$ is the victim set in the perturbed graph, but she does not know yet the isomorphism φ restricted to the set of victims $\{y_1, \ldots, y_m\}$. Lastly, the third condition states that $\delta/2$ is an upper bound on the distance between a victim's fingerprints in the pseudonymised graph φG^+ and the perturbed graph $t(\varphi G^+)$, where δ is the the minimum separation between the victim's fingerprints. In other words, the transformation method did not perturb a victim's fingerprint "too much". If those three conditions hold, Theorem 2 shows that the isomorphism φ restricted to the set of victims $\{y_1, \ldots, y_m\}$ is the trivial isomorphism onto $\{z_1, \ldots, z_m\}$.

Summing up: in this section we have enunciated the three problem formulations for robust active attacks, namely:

- Creating a robust set of fingerprints.
- Robustly retrieving the attacker subgraph in the perturbed graph.
- Robustly matching the original fingerprints to perturbed fingerprints.

Additionally, we have defined a set of conditions under which finding a solution for these problems guarantees a robust active attack to be successful. Each of the three enunciated problem has been stated as an optimisation task. Since obtaining exact solutions to these problems is computationally expensive, in the next section we introduce heuristics for finding approximate solutions.

5 Heuristics for an approximate instance of the robust active attack strategy

In this section we present the techniques for creating an instance of the robust active attack strategy described in the previous section. Since finding exact solutions to the optimisation problems in Equations (1), (2) and (3) is computationally expensive, we provide efficient approximate heuristics.

5.1 Attacker subgraph creation

For creating the internal links of the sybil subgraph, we will use the same strategy as the so-called *walk-based attack* (Backstrom et al., 2007), which is the most widely-studied instance of the original active attack strategy. By doing so, we make our new attack as (un)likely as the original to have the set of sybil nodes removed by sybil defences. Thus, for the set of sybil nodes S, the attack will set an arbitrary (but fixed) order among the elements of S. Let $x_1, x_2, \ldots, x_{|S|}$ represent

the vertices of S in that order. The attack will firstly create the path $x_1 x_2 \dots x_{|S|}$, whereas the remaining inter-sybil edges are independently created with probability 0.5.

For creating the set of fingerprints, we will apply a greedy algorithm for maximising the minimum separation defined in Equation (1). The idea behind the algorithm is to arrange all possible fingerprints in a grid-like auxiliary graph, in such a way that nodes representing similar fingerprints are linked by an edge, and nodes representing well-separated fingerprints are not. Looking for a set of maximally separated fingerprints in this graph reduces to a well-known problem in graph theory, namely that of finding an independent set. An *independent set* I of a graph G is a subset of vertices from G such that $E_{\langle I \rangle_G} = \emptyset$, that is, all vertices in I are pairwise not linked by edges. If the graph is constructed in such a way that every pair of fingerprints whose distance is less then or equal to some value i, then an independent set represents a set of fingerprints with a guaranteed minimum separation of at least $i + 1$. For example, the fingerprint graph shown in Figure 2 (a) represents the set of fingerprints $\{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$, with edges linking all pairs X, Y of fingerprints such that $d(X, Y) \leq 1$, whereas Figure 2 (b) represents an analogous graph where edges link all pairs X, Y of fingerprints such that $d(X, Y) \leq 2$. Note that the vertex set of both graphs is the power set of $\{1, 2, 3\}$, except for the empty set, which does not represent a valid fingerprint, as every victim must be linked to at least one sybil node. A set of fingerprints built from an independent set of the first graph may have minimum separation 2 (e.g. $\{\{1\}, \{2\}, \{1, 2, 3\}\}$) or 3 (e.g. $\{\{1, 3\}, \{2\}\}$), whereas a set of fingerprints built from an independent set of the second graph will have minimum separation 3 (the independent sets of this graph are $\{\{1\}, \{2, 3\}\}$, $\{\{1, 2\}, \{3\}\}$ and $\{\{1, 3\}, \{2\}\}$).

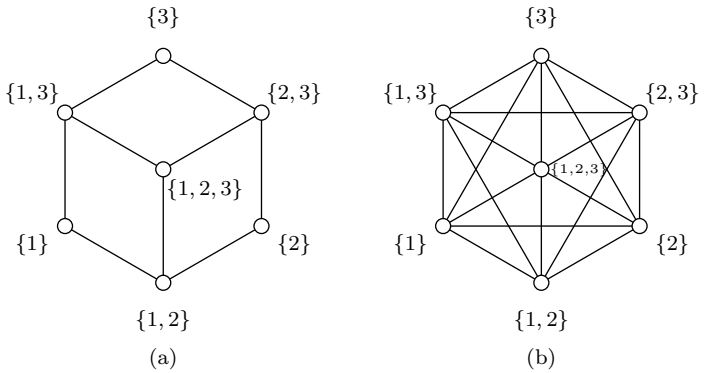


Fig. 2 The fingerprint graphs $(\mathcal{P}(\{1, 2, 3\}) \setminus \{\emptyset\}, \{(X, Y) \mid X, Y \in \mathcal{P}(\{1, 2, 3\}) \setminus \{\emptyset\}, X \neq Y, d(X, Y) \leq i\})$ for (a) $i = 1$ and (b) $i = 2$.

Our fingerprint generation method iteratively creates increasingly denser fingerprint graphs. The vertex set of every graph is the set of possible fingerprints, i.e. all subsets of S except the empty set. In the i-th graph, every pair of nodes X, Y such that $d(X, Y) \leq i$ will be linked by an edge. Thus, an independent set of this graph will be composed of nodes representing fingerprints whose minimum separation is at least $i+1$. A maximal[2] independent set of the fingerprint graph is computed in every iteration, to have an approximation of a maximum-cardinality set of uniformly distributed fingerprints with minimum separation at least $i + 1$. For example, in the graph of Figure 2 (a), the method will find $\{\{1\}, \{2\}, \{3\}, \{1, 2, 3\}\}$ as a maximum-cardinality set of uniformly distributed fingerprints with minimum separation 2; whereas for the graph of Figure 2 (b), the method will find, for instance, $\{\{1\}, \{2, 3\}\}$ as a maximum-cardinality set of uniformly distributed fingerprints with minimum separation 3. The method receives as a parameter a lower bound b on

[2] The maximum independent set problem is NP-hard, so it is infeasible to exactly compute a maximum independent set of every graph. Alternatively, we use a well known greedy approximation, which consists in iteratively finding a minimum-degree non-isolated vertex, and removing all its neighbours, until obtaining an empty graph, whose vertex set is an independent set of the original graph.

the number of fingerprints to generate. It iterates until the maximal independent set I_i obtained at the i-th step satisfies $|I_i| < b$, and gives I_{i-1} as output. Clearly, b must satisfy $b \geq m$, as every victim should be assigned a different fingerprint. Note that the algorithm does not guarantee to obtain exactly as many fingerprints as victims, so the output I_{i-1} is used as a pool, from which the attack randomly draws m fingerprints. Algorithm 1 lists the pseudo-code of this method.

Algorithm 1 Given a set S of sybil nodes and a positive integer b, compute a uniformly distributed set of fingerprints $F \subseteq \mathcal{P}(S)$ such that $|F| \geq b$.

$i \leftarrow 1$
$G_F^{(1)} \leftarrow (\mathcal{P}(S) \setminus \emptyset, \{(X,Y) \mid X,Y \in \mathcal{P}(S) \setminus \emptyset, X \neq Y, d(X,Y) \leq 1\})$
$I_1 \leftarrow \textsc{MaxIndSet}\left(G_F^{(1)}\right)$
repeat
  $F \leftarrow I_i$
  $i \leftarrow i + 1$
  $G_F^{(i)} \leftarrow (\mathcal{P}(S) \setminus \emptyset, \{(X,Y) \mid X,Y \in \mathcal{P}(S) \setminus \emptyset, X \neq Y, d(X,Y) \leq i\})$
  $I_i \leftarrow \textsc{MaxIndSet}\left(G_F^{(i)}\right)$
until $|I_i| < b$
return F

function $\textsc{MaxIndSet}(G = (V_G, E_G))$
  repeat
    $v \leftarrow \arg\min_{v \in V_G, \delta_G(v) \neq 0} \{\delta_G(v)\}$
    $E_G \leftarrow E_G \setminus \{(v,w) \mid w \in N_G(v)\}$
    $V_G \leftarrow V_G \setminus N_G(v)$
  until $E_G = \emptyset$
  return V_G
end function

In Algorithm 1, the order of every graph $G_F^{(i)}$ is $2^{|S|}-1$. Thus, the time complexity of every graph construction is $\mathcal{O}\left(\left(2^{|S|}\right)^2\right) = \mathcal{O}\left(2^{2|S|}\right)$. Moreover, the greedy algorithm for finding a maximal independent set runs in quadratic time with respect to the order of the graph, so in this case its time complexity is also $\mathcal{O}\left(2^{2|S|}\right)$. Finally, since the maximum possible distance between a pair of fingerprints is $|S|$, the worst-case time complexity of Algorithm 1 is $\mathcal{O}\left(|S| \cdot 2^{2|S|}\right)$. This worst case occurs when the number of victims is very small, as the number of times that steps 4 to 9 of the algorithm are repeated is more likely to approach $|S|$. While this time complexity may appear as excessive at first glance, we must consider that, for a social graph of order n, the algorithm will be run for sets of sybil nodes having at most cardinality $|S| = \log_2 n$. Thus, in terms of the order of the social graph, the worst-case running time will be $\mathcal{O}(n^2 \log_2 n)$.

5.2 Attacker subgraph retrieval

As discussed in Subsection 4.2, in the original formulation of active attacks, the sybil retrieval phase is based on the assumption that the attacker subgraph can be uniquely and exactly matched to a subgraph of the released graph. This assumption is relaxed by the formulation of robust attacker subgraph retrieval given in Definition 2, which accounts for the possibility that the attacker subgraph has been perturbed. The problem formulation given in Definition 2 requires a dissimilarity measure Δ to compare candidate subgraphs to the original attacker subgraph. We will introduce such a measure in this section. Moreover, the problem formulation requires searching the entire power set of $V_{t(\varphi G^+)}$, which is infeasible in practice. In order to reduce the size of the search space, we will establish a perturbation threshold ϑ, and the search procedure will discard any candidate subgraph X such that $\Delta(\langle X \rangle^w_{t(\varphi G^+)}, \langle S \rangle^w_{G^+}) > \vartheta$.

We now define the dissimilarity measure Δ that will be used. To that end, some additional notation will be necessary. For a graph H, a vertex set $V \subseteq V_H$, and a complete order $\prec \subseteq V \times V$,

we will define the vector $\mathbf{v}_{\prec} = (v_{i_1}, v_{i_2}, \ldots, v_{i_{|V|}})$, as the one satisfying $v_{i_1} \prec v_{i_2} \prec \ldots \prec v_{i_{|V|}}$. When the order $\prec$ is fixed or clear from the context, we will simply refer to $\mathbf{v}_{\prec}$ as $\mathbf{v}$. Moreover, for the sake of simplicity in presentation, we will in some cases abuse notation and use $\mathbf{v}$ for V, $\langle \mathbf{v} \rangle^w_H$ for $\langle V \rangle^w_H$, and so on. The search procedure assumes the existence of a fixed order $\prec_S$ on the original set of sybil nodes S, which is established at the attacker subgraph creation stage, as discussed in Subsection 5.1. In what follows, we will use the notation $\mathbf{s} = (x_1, x_2, \ldots, x_{|S|})$ for the vector $\mathbf{s}_{\prec_S}$.

Given the original attacker subgraph $\langle S \rangle^w_{G^+}$ and a subgraph of $t(\varphi G^+)$ weakly induced by a candidate vector $\mathbf{v} = (v_1, v_2, \ldots, v_{|S|})$, the dissimilarity measure Δ will compare $\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}$ to $\langle S \rangle^w_{G^+}$ according to the following criteria:

- The set of *inter-sybil edges* of $\langle S \rangle^w_{G^+}$ will be compared to that of $\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}$. This is equivalent to comparing $E(\langle S \rangle_{G^+})$ and $E(\langle \mathbf{v} \rangle_{t(\varphi G^+)})$. To that end, we will apply to $\langle S \rangle_{G^+}$ the isomorphism $\varphi' \colon \langle S \rangle_{G^+} \to \langle \mathbf{v} \rangle_{t(\varphi G^+)}$, which makes $\varphi'(x_i) = v_i$ for every $i \in \{1, \ldots, |S|\}$. The contribution of inter-sybil edges to Δ will thus be defined as

$$\Delta_{syb}\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) = \left| E(\varphi' \langle S \rangle_{G^+}) \nabla E(\langle \mathbf{v} \rangle_{t(\varphi G^+)}) \right|, \tag{4}$$

  that is, the symmetric difference between the edge sets of $\varphi' \langle S \rangle_{G^+}$ and $\langle \mathbf{v} \rangle_{t(\varphi G^+)}$.
- The set of *sybil-to-non-sybil edges* of $\langle S \rangle^w_{G^+}$ will be compared to that of $\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}$. Unlike the previous case, where the orders $\prec_S$ and $\prec_{\mathbf{v}}$ allow to define a trivial isomorphism between the induced subgraphs, in this case creating the appropriate matching would be equivalent to solving the re-identification problem for every candidate subgraph, which is considerably inefficient. In consequence, we introduce a relaxed criterion, which is based on the numbers of non-sybil neighbours of every sybil node, which we refer to as *marginal degrees*. The marginal degree of a sybil node $x \in S$ is thus defined as $\delta'_{\langle S \rangle^w_{G^+}}(x) = \left| N_{\langle S \rangle^w_{G^+}}(x) \setminus S \right|$. By analogy, for a vertex $v \in \mathbf{v}$, we define $\delta'_{\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}}(v) = \left| N_{\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}}(v) \setminus \mathbf{v} \right|$. Finally, the contribution of sybil-to-non-sybil edges to Δ will be defined as

$$\Delta_{neigh}\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) = \sum_{i=1}^{|S|} \left| \delta'_{\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}}(v_i) - \delta'_{\langle S \rangle^w_{G^+}}(x_i) \right| \tag{5}$$

- The dissimilarity measure combines the previous criteria as follows:

$$\Delta\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) = \Delta_{syb}\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) + \Delta_{neigh}\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) \tag{6}$$

Figure 3 shows an example of the computation of this dissimilarity measure, with $\mathbf{s} = (x_1, x_2, x_3, x_4, x_5)$ and $\mathbf{v} = (v_1, v_2, v_3, v_4, v_5)$. In the figure, we can observe that $(x_1, x_3) \in E_{\langle S \rangle^w_{G^+}}$ and $(v_1, v_3) \notin E_{\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}}$, whereas $(x_3, x_4) \in E_{\langle S \rangle^w_{G^+}}$ and $(v_3, v_4) \notin E_{\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}}$. In consequence, $\Delta_{syb}\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) = 2$. Moreover, we can also observe that $\delta'_{\langle S \rangle^w_{G^+}}(x_2) = |\emptyset| = 0$, whereas $\delta'_{\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}}(v_2) = |\{y'_1, y'_5\}| = 2$. Since $\delta'_{\langle S \rangle^w_{G^+}}(x_i) = \delta'_{\langle \mathbf{v} \rangle^w_{t(\varphi G^+)}}(v_i)$ for $i \in \{1, 3, 4, 5\}$, we have $\Delta_{neigh}\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) = 2$, so $\Delta\left(\langle S \rangle^w_{G^+}, \langle \mathbf{v} \rangle^w_{t(\varphi G^+)}\right) = 4$. It is simple to see that the value of the dissimilarity function is dependent on the order imposed by the vector $\mathbf{v}$. For example, consider

the vector $\mathbf{v}' = (v_5, v_2, v_3, v_4, v_1)$. We can verify[3] that now $\Delta_{syb}\left(\langle S\rangle^w_{G^+}, \langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}\right) = 4$, whereas $\Delta_{neigh}\left(\langle S\rangle^w_{G^+}, \langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}\right) = 4$, so the dissimilarity value is now $\Delta\left(\langle S\rangle^w_{G^+}, \langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}\right) = 8$.

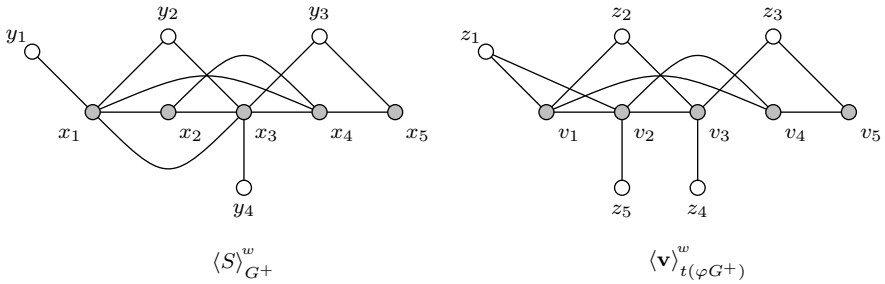


Fig. 3 An example of possible graphs $\langle S\rangle^w_{G^+}$ and $\langle \mathbf{v}\rangle^w_{t(\varphi G^+)}$. Vertices in S and $\mathbf{v}$ are coloured in gray.

The search procedure assumes that the transformation t did not remove the image of any sybil node from φG^+, so it searches the set of cardinality-$|S|$ permutations of elements from $V_{t(\varphi G^+)}$, respecting the tolerance threshold. The method is a breadth-first search, which analyses at the i-th level the possible matches to the vector $(x_1, x_2, \ldots, x_i)$ composed of the first i components of $\mathbf{s}$. The tolerance threshold ϑ is used to prune the search tree. A detailed description of the procedure is shown in Algorithm 2. Ideally, the algorithm outputs a unitary set $\tilde{\mathcal{C}}^* = \{(v_{j_1}, v_{j_2}, \ldots, v_{j_{|S|}})\}$, in which case the vector $\mathbf{v} = (v_{j_1}, v_{j_2}, \ldots, v_{j_{|S|}})$ is used as the input to the fingerprint matching phase, described in the following subsection. If this is not the case, and the algorithm yields $\tilde{\mathcal{C}}^* = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t\}$, the attack randomly picks an element $\mathbf{v}_i \in \tilde{\mathcal{C}}^*$ and proceeds to the fingerprint matching phase. Finally, if $\tilde{\mathcal{C}}^* = \emptyset$, the attack is considered to fail, as no re-identification is possible.

To conclude the discussion of Algorithm 2, we point out that if it is executed with $\vartheta = 0$, then $\tilde{\mathcal{C}}^*$ contains exactly the same candidate set that would be recovered by the attacker subgraph retrieval phase of the original walk-based attack. In fact, in choosing a value for the parameter ϑ, the practitioner must assess the trade-off between the retrieval capability of the attack and its computational cost. On the one hand, a low tolerance threshold leads to a fast execution of the retrieval method, at the cost of a higher risk of failing to retrieve a largely perturbed sybil subgraph. On the other hand, by making the tolerance threshold arbitrarily large, one can guarantee that the sybil subgraph will not be discarded during the search process[4]. However, in this case the retrieval method may end-up performing a near-to-exhaustive search, which is prohibitively expensive in terms of memory and execution time.

5.3 Fingerprint matching

Now, we describe the noise-tolerant fingerprint matching process. Let $Y = \{y_1, \ldots, y_m\} \subseteq V_{G^+}$ represent the set of victims. Let S be the original set of sybil nodes and $\tilde{S}' \subseteq V_{t(\varphi G^+)}$ a candidate obtained by the sybil retrieval procedure described above. As in the previous subsection, let $\mathbf{s} =$

[3] We now have that $(x_1, x_2) \in E_{\langle S\rangle^w_{G^+}}$ and $(v_5, v_2) \notin E_{\langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}}$; $(x_1, x_3) \in E_{\langle S\rangle^w_{G^+}}$ and $(v_5, v_3) \notin E_{\langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}}$; $(x_2, x_5) \notin E_{\langle S\rangle^w_{G^+}}$ and $(v_2, v_1) \in E_{\langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}}$; and $(x_3, x_4) \in E_{\langle S\rangle^w_{G^+}}$ whereas $(v_3, v_4) \notin E_{\langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}}$. Moreover, now $\left|\delta'_{\langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}}(v_5) - \delta'_{\langle S\rangle^w_{G^+}}(x_1)\right| = 1$, $\left|\delta'_{\langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}}(v_2) - \delta'_{\langle S\rangle^w_{G^+}}(x_2)\right| = 2$ and $\left|\delta'_{\langle \mathbf{v}'\rangle^w_{t(\varphi G^+)}}(v_1) - \delta'_{\langle S\rangle^w_{G^+}}(x_5)\right| = 1$.

[4] Note that it is still possible that the sybil subgraph is excluded from the final output. This would be the case if, after perturbation, some other subgraph (a false positive) happens to be more similar to the original sybil subgraph.

Algorithm 2 Given the graphs G^+ and $t(\varphi G^+)$, the set of original sybil nodes $S \subseteq V_{G^+}$, and the maximum distance threshold ϑ, obtain the set $\tilde{\mathcal{C}}^*$ of equally-likely best candidate sybil sets.

▷ Find suitable candidates to match x_1
$PartialCandidates_1 \leftarrow \emptyset$
$d \leftarrow \vartheta$
for $v \in V_{t(\varphi G^+)}$ **do**
  if $\Delta\left(\langle(x_1)\rangle^w_{G^+}, \langle(v)\rangle^w_{t(\varphi G^+)}\right) < d$ **then**
    $PartialCandidates_1 \leftarrow \{(v)\}$
    $d \leftarrow \Delta\left(\langle(x_1)\rangle^w_{G^+}, \langle(v)\rangle^w_{t(\varphi G^+)}\right)$
  else if $\Delta\left(\langle(x_1)\rangle^w_{G^+}, \langle(v)\rangle^w_{t(\varphi G^+)}\right) = d$ **then**
    $PartialCandidates_1 \leftarrow PartialCandidates_1 \cup \{(v)\}$
  end if
end for
if $PartialCandidates_1 = \emptyset$ **then**
  return $\emptyset$
else if $|S| = 1$ **then**
  return $PartialCandidates_1$
else
  ▷ Find rest of matches for candidates
  return Breadth-First-Search(2, $PartialCandidates_1$)
end if

---

function Breadth-First-Search(i, $PartialCandidates_{i-1}$)
  ▷ Find suitable candidates to match $(x_1, x_2, \ldots, x_i)$
  $\mathbf{s}' \leftarrow (x_1, x_2, \ldots, x_i)$
  $PartialCandidates_i \leftarrow \emptyset$
  $d \leftarrow \vartheta$
  for $(v_{j_1}, v_{j_2}, \ldots, v_{j_{i-1}}) \in \mathcal{C}_{i-1}$ **do**
    $ExtendedCandidates \leftarrow \emptyset$
    $d' \leftarrow \vartheta$
    for $w \in V_{t(\varphi G^+)} \setminus (v_{j_1}, v_{j_2}, \ldots, v_{j_{i-1}})$ **do**
      $\mathbf{v}' \leftarrow (v_{j_1}, v_{j_2}, \ldots, v_{j_{i-1}}, w)$
      if $\Delta\left(\langle\mathbf{s}'\rangle^w_{G^+}, \langle\mathbf{v}'\rangle^w_{t(\varphi G^+)}\right) < d'$ **then**
        $ExtendedCandidates \leftarrow \{\mathbf{v}'\}$
        $d' \leftarrow \Delta\left(\langle\mathbf{s}'\rangle^w_{G^+}, \langle\mathbf{v}'\rangle^w_{t(\varphi G^+)}\right)$
      else if $\Delta\left(\langle\mathbf{s}'\rangle^w_{G^+}, \langle\mathbf{v}'\rangle^w_{t(\varphi G^+)}\right) = d'$ **then**
        $ExtendedCandidates \leftarrow ExtendedCandidates \cup \{\mathbf{v}'\}$
      end if
    end for
    if $d' < d$ **then**
      $PartialCandidates_i \leftarrow ExtendedCandidates$
      $d \leftarrow d'$
    else if $d' = d$ **then**
      $PartialCandidates_i \leftarrow PartialCandidates_i \cup ExtendedCandidates$
    end if
  end for
  if $PartialCandidates_i = \emptyset$ **then**
    return $\emptyset$
  else if $i = |S|$ **then**
    return $PartialCandidates_i$
  else
    ▷ Find rest of matches for candidates
    return Breadth-First-Search($i + 1$, $PartialCandidates_i$)
  end if
end function

$(x_1, x_2, \ldots, x_{|S|})$ be the vector containing the elements of S in the order imposed at the sybil subgraph creation stage. Moreover, let $\langle \mathbf{v} \rangle^{w}_{t(\varphi G^+)}$, with $\mathbf{v} = (v_1, v_2, \ldots, v_{|S|}) \in \tilde{\mathcal{C}}^*$, be a candidate sybil subgraph, retrieved using Algorithm 2. Finally, for every $i \in \{1, \ldots, m\}$, let $F_i \subseteq S$ be the original fingerprint of the victim y_i and $\phi F_i \subseteq \mathbf{v}$ its image by the isomorphism mapping $\mathbf{s}$ to $\mathbf{v}$.

We now describe the process for finding $Y' = \{y'_1, \ldots, y'_m\} \subseteq V_{t(\varphi G^+)}$, where $y'_i = \varphi(y_i)$, using $\phi F_1, \phi F_2, \ldots, \phi F_m$, $\mathbf{s}$ and $\mathbf{v}$. If the perturbation $t(G^+)$ had caused no damage to the fingerprints, checking for the exact matches is sufficient. Since, as previously discussed, this is usually not the case, we will introduce a noise-tolerant fingerprint matching strategy that maps every original fingerprint to its most similar candidate fingerprint, within some tolerance threshold β.

Algorithm 3 describes the process for finding the set of optimal re-identifications. For a candidate victim $z \in N_{t(\varphi G^+)}(\mathbf{v}) \setminus \mathbf{v}$, the algorithm denotes as $\tilde{F}_{z,\mathbf{v}} = N_{t(\varphi G^+)}(z) \cap \mathbf{v}$ its fingerprint with respect to $\mathbf{v}$. The algorithm is a depth-first search procedure. First, the algorithm finds, for every ϕF_i, $i \in \{1, \ldots, m\}$, the set of most similar candidate fingerprints, and keeps the set of matches that reach the minimum distance. From these best matches, one or several partial re-identifications are obtained. The reason why more than one partial re-identification is obtained is that more than one candidate fingerprint may be equally similar to some ϕF_i. For every partial re-identification, the method recursively finds the set of best completions and combines them to construct the final set of equally likely re-identifications. The search space is reduced by discarding insufficiently similar matches. For any candidate victim z and any original victim y_i such that $d(\tilde{F}_{z,\mathbf{v}}, \phi F_i) < \beta$, the algorithm discards all matchings where $\varphi(y_i) = z$.

To illustrate how the method works, recall the graphs $\langle S \rangle^{w}_{G^+}$ and $\langle \mathbf{v} \rangle^{w}_{t(\varphi G^+)}$ depicted in Figure 3. The original set of victims is $Y = \{y_1, y_2, y_3, y_4\}$ and their fingerprints are $F_1 = \{x_1\}$, $F_2 = \{x_1, x_3\}$, $F_3 = \{x_3, x_5\}$, and $F_4 = \{x_3\}$, respectively. In consequence, we have $\phi F_1 = \{v_1\}$, $\phi F_2 = \{v_1, v_3\}$, $\phi F_3 = \{v_3, v_5\}$, and $\phi F_4 = \{v_3\}$. The set of candidate victims is $N_{t(\varphi G^+)}(\mathbf{v}) \setminus \mathbf{v} = \{z_1, z_2, z_3, z_4, z_5\}$. The method will first find all exact matchings, that is $\varphi(y_2) = z_2$, $\varphi(y_3) = z_3$, and $\varphi(y_4) = z_4$, because the distances between the corresponding fingerprints is zero in all three cases. Since none of these matchings is ambiguous, the method next determines the match $\varphi(y_1) = z_1$, because $d(\tilde{F}_{z_1}, \phi F_1) = d(\{v_1, v_2\}, \{v_1\}) = 1 < 2 = d(\{v_2\}, \{v_1\}) = d(\tilde{F}_{z_5}, \phi F_1)$. At this point, the method stops and yields the unique re-identification $\{(y_1, z_1), (y_2, z_2), (y_3, z_3), (y_4, z_4)\}$. Now, suppose that the vertex z_5 is linked in $t(\varphi G^+)$ to v_3, instead of v_2, as depicted in Figure 4. In this case, the method will unambiguously determine the matchings $\varphi(y_2) = z_2$ and $\varphi(y_3) = z_3$, and then will try the two choices $\varphi(y_4) = z_4$ and $\varphi(y_4) = z_5$. In the first case, the method will make $\varphi(y_1) = z_1$ and discard z_5. Analogously, in the second case the method will also make $\varphi(y_1) = z_1$, and will discard z_4. Thus, the final result will consist in two equally likely re-identifications, namely $\{(y_1, z_1), (y_2, z_2), (y_3, z_3), (y_4, z_4)\}$ and $\{(y_1, z_1), (y_2, z_2), (y_3, z_3), (y_4, z_5)\}$.

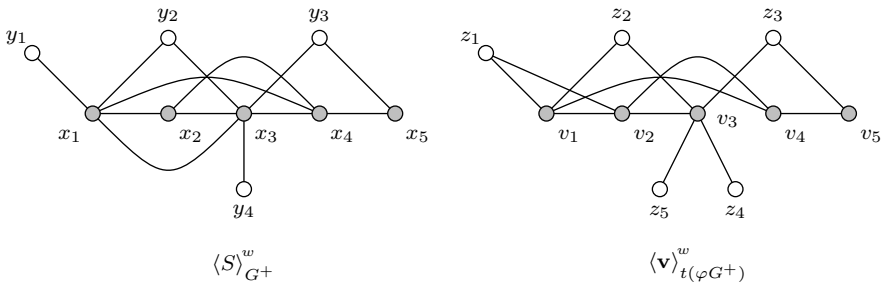


Fig. 4 Alternative example of possible graphs $\langle S \rangle^{w}_{G^+}$ and $\langle \mathbf{v} \rangle^{w}_{t(\varphi G^+)}$.

Ideally, Algorithms 2 and 3 both yield unique solutions, in which case the sole element in the output of Algorithm 3 is given as the final re-identification. If this is not the case, the attack picks a random candidate sybil subgraph from $\tilde{\mathcal{C}}^*$, uses it as the input of Algorithm 3, and picks a random re-identification from its output. If either algorithm yields an empty solution, the attack

Algorithm 3 Given the graphs G^+ and $t(\varphi G^+)$, the original set of victims $Y = \{y_1, y_2, \ldots, y_m\}$, the original fingerprints $F_1, F_2, \ldots, F_m$, a candidate set of sybils $\mathbf{v}$, and the maximum distance threshold β, obtain the set $ReIdents$ of best matchings.

$(ReIdents, d) \leftarrow$ GreedyMatching(Y, $N_{t(\varphi G^+)}(\mathbf{v}) \setminus \mathbf{v}$)
return $ReIdents$

function GreedyMatching(Y, M)
  ▷ Find best matches of some unmapped victim(s) to one or more candidate victims
  $MapPartialBest \leftarrow \emptyset$
  $d \leftarrow \beta$
  for $y_i \in Y$ **do**
    for $z \in M$ **do**
      if $d(\tilde{F}_{z,\mathbf{v}}, \phi F_i) < d$ **then**
        $MapPartialBest \leftarrow \{(y_i, \{z\})\}$
        $d \leftarrow d(\tilde{F}_{z,\mathbf{v}}, \phi F_i)$
      else if $d(\tilde{F}_{z,\mathbf{v}}, \phi F_i) = d$ **then**
        if $(y_i, P) \in MapPartialBest$ **then**
          $MapPartialBest \leftarrow (MapPartialBest \setminus (y_i, P)) \cup \{(y_i, P \cup \{z\})\}$
        else
          $MapPartialBest \leftarrow MapPartialBest \cup \{(y_i, P \cup \{z\})\}$
        end if
      end if
    end for
  end for
  if $MapPartialBest = \emptyset$ **then**
    return $(\emptyset, +\infty)$
  else
    ▷ Build partial re-identifications from best matches
    Pick any $(y, P) \in MapPartialBest$
    $PartialReIdents \leftarrow \{(y, z) \mid z \in P\}$
    for $(y', P') \in MapPartialBest \setminus (y, P)$ **do**
      $PartialReIdents \leftarrow \{R \cup \{y', z'\} \mid R \in PartialReIdents \wedge z' \in P'\}$
    end for
    if $|Y| = 1$ **then**
      return $(PartialReIdents, d)$
    else
      ▷ Recursively call the method to find best completions for partial re-identifications
      $BestComplReIdent \leftarrow \emptyset$
      $d_{best} \leftarrow \beta$
      for $R \in PartialReIdents$ **do**
        $(CompletedReIdents, d) \leftarrow$ GreedyMatching($Y \setminus \{y \mid (y, z) \in R\}$, $M \setminus \{z \mid (y, z) \in R\}$)
        if $d < d_{best}$ **then**
          $BestComplReIdent \leftarrow CompletedReIdents$
          $d_{best} \leftarrow d$
        else if $d = d_{best}$ **then**
          $BestComplReIdent \leftarrow BestComplReIdent \cup CompletedReIdents$
        end if
      end for
      return $(\{P \cup R \mid P \in PartialReIdents \wedge R \in BestComplReIdent\}, d_{best})$
    end if
  end if
end function

fails. Finally, it is important to note that, if Algorithm 2 is run with $\vartheta = 0$ and Algorithm 3 is run with $\beta = 0$, then the final result is exactly the same set of equally likely matchings that would be obtained by the original walk-based attack. As for the case of sybil subgraph retrieval, in choosing the value of the parameter β for robust fingerprint matching, the practitioner must consider the trade-off between efficiency and noise tolerance, in a manner analogous to the one discussed for the selection of the parameter ϑ in Algorithm 2.

6 Experiments

The purpose of our experiments[5] is threefold. Firstly, we show the considerable gain in resilience of robust active attacks, in comparison to the original walk-based attack. Secondly, we assess the contributions of different components of the robust active attack strategy to the success of the attacks. Finally, we analyse the weaknesses shared by the robust and the original active attacks, and discuss how they may serve as the basis for the development of new privacy-preserving graph publication methods. Each run of our experiments is determined by the the selection of a graph type, a perturbation method and an active attack strategy. In what follows, we first describe the choices for each of these three components, and conclude the section by discussing the empirical results obtained.

6.1 Three models of synthetic graphs and two real-life networks

In order to make the results reported in this section comparable to those reported for the walk-based attack on anonymised graphs (Mauw et al., 2018b), we first study the behaviour of the attacks under evaluation on Erdős-Rényi (ER) random graphs (Erdős and Rényi, 1959) of order 200. We generated 200,000 ER graphs, 10,000 featuring every density value in the set $\{0.05, 0.1, \ldots, 1.0\}$.

We also study the behaviour of the attacks on Watts-Strogatz (WS) small world graphs (Watts and Strogatz, 1998) and Barabási-Albert (BA) scale-free graphs (Barabási and Albert, 1999). The WS model has two parameters, the number K of neighbours originally assigned to every vertex, and the probability ϱ that an edge of the initial K-regular ring lattice is randomly rewritten. In our experiments, we generated 10,000 graphs of order 200 for every pair (K, ϱ), where $K \in \{10, 20, \ldots, 100\}$ and $\varrho \in \{0.25, 0.5, 0.75\}$. In the case of BA graphs, we used seed graphs of order 50 and every graph was grown by adding 150 vertices, and performing the corresponding edge additions. The BA model has a parameter m defining the number of new edges added for every new vertex. Here, we generated 10,000 graphs for every value of m in the set $\{5, 10, \ldots, 50\}$. In generating every graph, the seed graph was chosen to be, with probability $1/3$, a complete graph, an m-regular ring lattice, or an ER random graph of density 0.5.

Finally, in order to complement the results obtained on randomly generated graphs, and to showcase the behaviour of the attacks in larger, real-life social networks, we additionally study the attacks in the context of two benchmark social graphs. The first one, which is commonly referred to as the *Panzarasa graph*, after one of its creators (Panzarasa et al., 2009), was collected from an online community of students at the University of California, Irvine. In the Panzarasa graph, a directed edge (A, B) represents that student A sent at least one message to student B. In our experiments, we use a processed version of this graph, where edge orientation, loops and isolated vertices were removed. This graph has 1,893 vertices and 20,296 edges. The second real-life social graph that we use was constructed from a collection of e-mail messages exchanged between students, professors and staff at Universitat Rovira i Virgili (URV), Spain (Guimera et al., 2003). For the construction of the graph, the data collectors added an edge between every pair of users that messaged each other. In doing so, they ignored group messages with more than 50 recipients. Moreover, they removed isolated vertices and connected components of order 2. The URV graph has 1,133 vertices and 5,451 edges.

[5] We performed our experiments on the HPC platform of the University of Luxembourg (Varrette et al., 2014). In particular, we ran our experiments on the Gaia cluster of the UL HPC. A detailed description of the Gaia cluster is available at https://hpc.uni.lu/systems/gaia/. Leveraging the high level of parallelism achievable on that platform, we obtained the results discussed in this section in approximately 240 hours for each graph model. The implementations of the graph generators, anonymisation methods and attack simulations are available at https://github.com/rolandotr/graph.

6.2 Graph perturbation

In our experiments we considered existing anonymisation methods against active attacks and random perturbation. Although the latter does not provide formal privacy guarantees, it is a useful benchmark for evaluating resilience against noisy releases. The graph perturbation methods used in our experiments are the following:

(a) An algorithm enforcing (k, ℓ)-anonymity for some $k > 1$ or some $\ell > 1$ (Mauw et al., 2016).
(b) An algorithm enforcing $(2, \Gamma_{G,1})$-anonymity (Mauw et al., 2018b).
(c) An algorithm enforcing $(k, \Gamma_{G,1})$-adjacency anonymity for a given value of k (Mauw et al., 2018b). Here, we run the method with $k = |S|$ (*i.e.* the number of sybil nodes), since it has been empirically shown that the original walk-based attack is very likely to be thwarted in this case (Mauw et al., 2018b).
(d) Randomly flipping 1% of the edges in G^+. Each flip consists in randomly selecting a pair of vertices $u, v \in V_{G^+}$, removing the edge (u, v) if it belongs to E_{G^+}, or adding it in the opposite case. In the case of synthetic graphs, since every instance of G^+ has order $n = 208$, this perturbation performs $\left\lfloor 0.01 \cdot \frac{n(n-1)}{2} \right\rfloor = 215$ flips. In the case of the Panzarasa graph, the order of G^+ is $1,904$, so this perturbation performs $18,116$ flips. Finally, in the case of the URV graph, the order of G^+ is $1,144$, so this perturbation performs $6,537$ flips.
(e) Randomly flipping 5% of the edges in G^+ (that is $1,076$ flips on synthetic graphs, $90,582$ on the Panzarasa graph and $32,689$ on the URV graph), in a manner analogous to the one used above.
(f) Randomly flipping 10% of the edges in G^+ (that is $2,153$ flips on synthetic graphs, $181,165$ on the Panzarasa graph and $65,379$ on the URV graph), in a manner analogous to the one used above.

6.3 Attack variants

We compare the behaviour of the original walk-based attack to four instantiations of the robust attack described in Section 5. All four instantiations have in common the use of noise-tolerant sybil subgraph retrieval and fingerprint matching, since noise tolerance is the basis of the notion of robustness of the new attacks. The differences between the instances are given by the combination of choices in terms of two features: 1) the use of high versus low noise-tolerance thresholds, and 2) the use of maximally separated fingerprints versus the use of randomly generated fingerprints. In both cases, each choice represents the alternative between attacks featuring higher robustness at the cost of an overhead in computation (higher noise tolerance, maximally separated fingerprints) and attacks which are more efficient but sacrifice some robustness features (lower noise tolerance, randomly generated fingerprints). Table 1 summarises the list of attack variants to be compared. Note that the attacks labelled as Robust-High-Max and Robust-Low-Max in Table 1 are both instances of the robust active attack featuring all its components, and only differ in the tolerance levels used in sybil subgraph retrieval and fingerprint matching. Also note that attack variants using maximally separated fingerprints are run with a random component as well, because a set of maximally separated fingerprints may be larger than the number of victims and in this case the fingerprints used for every run of the attacks are randomly selected from this pool.

As discussed by Backstrom et al. (2007), for a graph of order n, it suffices to insert $\log_2 n$ sybil nodes for being able to compromise any possible victim, whereas even the so-called *near-optimal* sybil defences (Yu et al., 2006, 2008) do not aim to remove every sybil node, but to limit their number to around $\log_2 n$. In light of these two considerations, when evaluating every attack variant on the collections of synthetic graphs, we use 8 sybil nodes, as $\lceil \log_2 200 \rceil = 8$. For the same reason, we use 11 sybil nodes on the Panzarasa and URV graphs. In all cases, we use the same number of victims as sybil nodes, to make the results comparable to those reported for the walk-based attack on anonymised graphs (Mauw et al., 2018b).

Attack id	Noise-tolerant sybil subgraph retrieval and fingerprint matching	Tolerance threshold levels	Maximally separated fingerprints
Original	No	Not applicable	Not applicable
Robust-Low-Rand	Yes	Low	No
Robust-High-Rand	Yes	High	No
Robust-Low-Max	Yes	Low	Yes
Robust-High-Max	Yes	High	Yes

Table 1 Descriptions of the attacks to be compared.

Finally, we set the tolerance thresholds to be $\vartheta = \beta = 8$ for the attacks Robust-High-Rand and Robust-High-Max, and $\vartheta = \beta = 4$ for Robust-Low-Rand and Robust-Low-Max, when executed on synthetic graphs. In the case of the Panzarasa and URV graphs, due to their considerably larger size, and in order to keep the execution time and memory consumption of Algorithms 2 and 3 within reasonable limits, we set the tolerance thresholds to be $\vartheta = \beta = 4$ for the attacks Robust-High-Rand and Robust-High-Max, and $\vartheta = \beta = 2$ for Robust-Low-Rand and Robust-Low-Max.

6.4 Probability of success of the attacks

Following the attacker-defender game in Section 3, for every graph and every attack variant in Table 1, we first run the attacker subgraph creation stage. Then, for every resulting graph, we obtain six variants of anonymised graphs, which differ from each other in the perturbation method applied (items (a) to (f) listed in Section 6.2). Finally, for each perturbed graph, we simulate the execution of the re-identification stage and compute its probability of success as follows:

$$\Pr = \begin{cases} \frac{\sum_{X \in \mathcal{X}} p_X}{|\mathcal{X}|} & \text{if } \mathcal{X} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\mathcal{X}$ is the set of equally-likely possible sybil subgraphs retrieved in $t(\varphi G^+)$ by the third phase of the attack, and

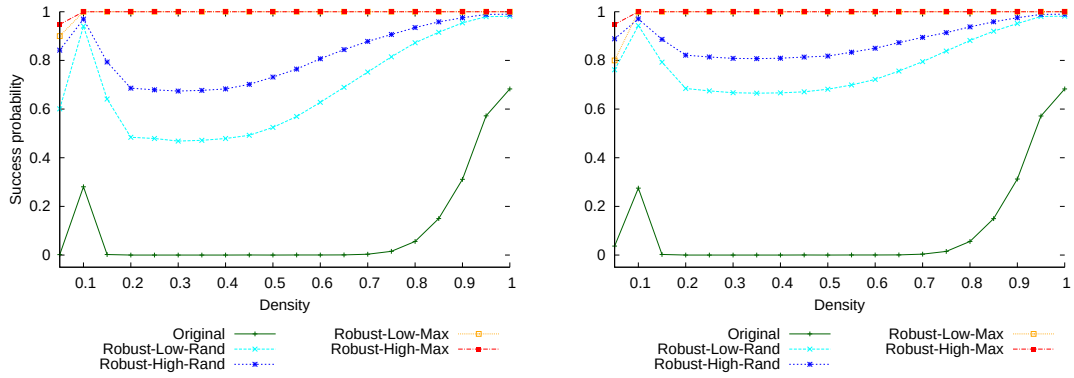
$$p_X = \begin{cases} \frac{1}{|\mathcal{Y}_X|} & \text{if} \quad Y \in \mathcal{Y}_X \\ 0 & \text{otherwise} \end{cases}$$

with $\mathcal{Y}_X$ containing all equally-likely fingerprint matchings according to X. Note that, for the original walk-based attack, $\mathcal{X}$ is the set of subgraphs of $t(\varphi G^+)$ isomorphic to $\langle S \rangle^w_{G^+}$, whereas for the robust attack we have $\mathcal{X} = \left\{ \langle \mathbf{v} \rangle^w_{t(\varphi G^+)} \mid \mathbf{v} \in \tilde{\mathcal{C}}^* \right\}$, being $\tilde{\mathcal{C}}^*$ the output of Algorithm 2. Moreover, for the original walk-based attack, $\mathcal{Y}_X = \{\{y_1, y_2 \ldots, y_m\} \subseteq V_{t(\varphi G^+)} \mid \forall i \in \{1, \ldots, m\} \colon F_{y_i, X} = F_i\}$, whereas for the robust attack $\mathcal{Y}_X$ is the output of Algorithm 3.

In the case of synthetic graphs, in order to obtain the scores used for comparing the different approaches, we computed, for every combination of an attack variant and a perturbation strategy, the average of the success probabilities over every group of 10,000 graphs sharing the same set of parameter choices (as described in Subsection 6.1). In the case of the two real networks, we executed, for every combination of an attack variant and a perturbation strategy, 10 runs on the Panzarasa graph and 10 runs on the URV graph. In each of these runs, a different set of victims was randomly chosen. The final scores used for comparisons were the averaged success probabilities over every group of runs.

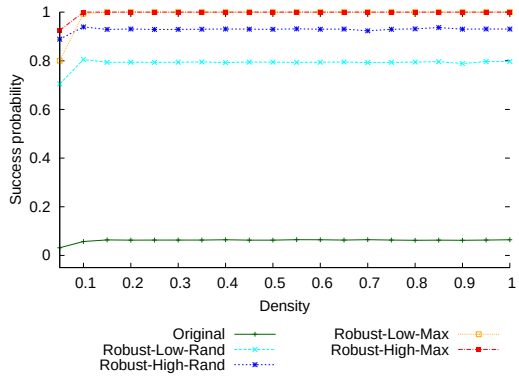
6.5 Analysis of results

Figure 5 shows the averaged success probabilities of the five attack variants on the set of Erdős-Rényi random graphs, after applying the perturbation strategies (a) to (f). Every chart represents

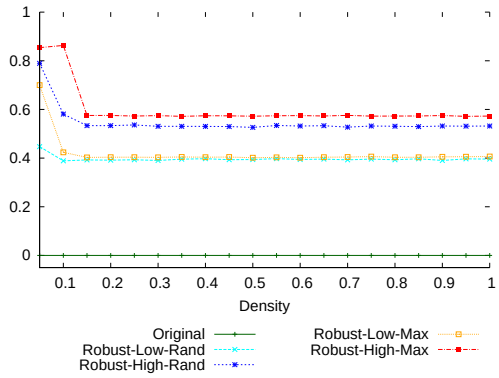


(a) (k, ℓ)-anonymity, $k > 1$ or $\ell > 1$

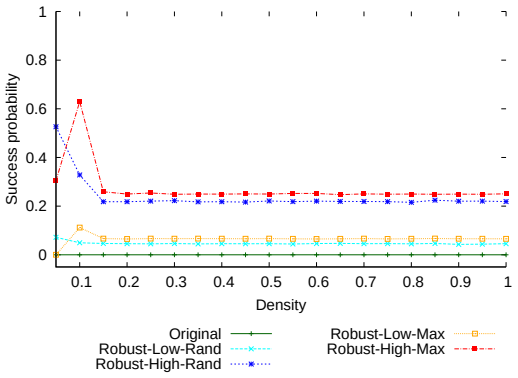
(b) $(2, \Gamma_{G,1})$-anonymity



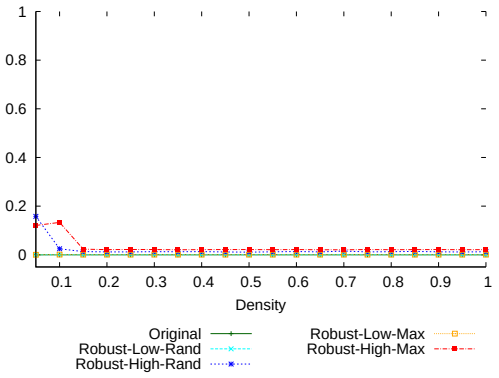
(c) $(|S|, \Gamma_{G,1})$-adjacency anonymity



(d) 1% random perturbation



(e) 5% random perturbation

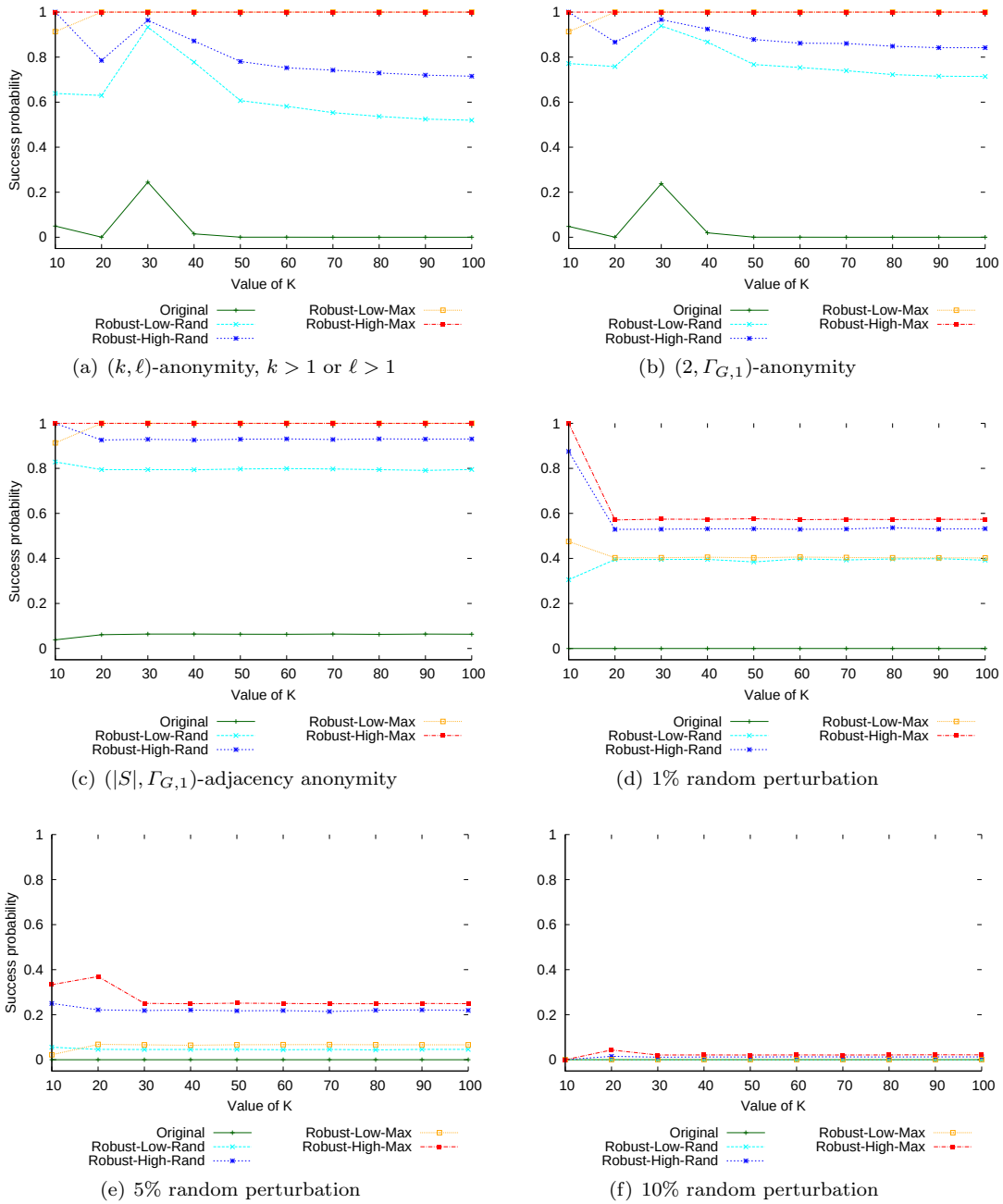


(f) 10% random perturbation

Fig. 5 Success probabilities of every attack variant on the collection of Erdős-Rényi random graphs, after publishing the graphs perturbed by the methods listed above.

the behaviour of the attacks on graphs perturbed with a specific method. The x axis displays density values and the y axis displays success probabilities. Analogous results are shown in Figures 6, 7 and 8 for Watts-Strogatz random graphs with $\varrho = 0.25$, $\varrho = 0.5$ and $\varrho = 0.75$, respectively. In this case, in every chart the x axis displays the values of K. Likewise, Figure 9 shows analogous results for Barabási-Albert random graphs. Here, in every chart the x axis displays the values of m. Finally, Table 2 shows the averaged success probabilities of the five attack variants on the two real-life networks. In the table, every row represents the combination of a network and

a perturbation method, and every column represents an attack variant. Success probabilities are rounded to four significant figures.



(a) (k, ℓ)-anonymity, $k > 1$ or $\ell > 1$

(b) $(2, \Gamma_{G,1})$-anonymity

(c) $(|S|, \Gamma_{G,1})$-adjacency anonymity

(d) 1% random perturbation

(e) 5% random perturbation

(f) 10% random perturbation

Fig. 6 Success probabilities of every attack variant on the collection of Watts-Strogatz small-world random graphs, with $\varrho = 0.25$, after publishing the graphs perturbed by the methods listed above.

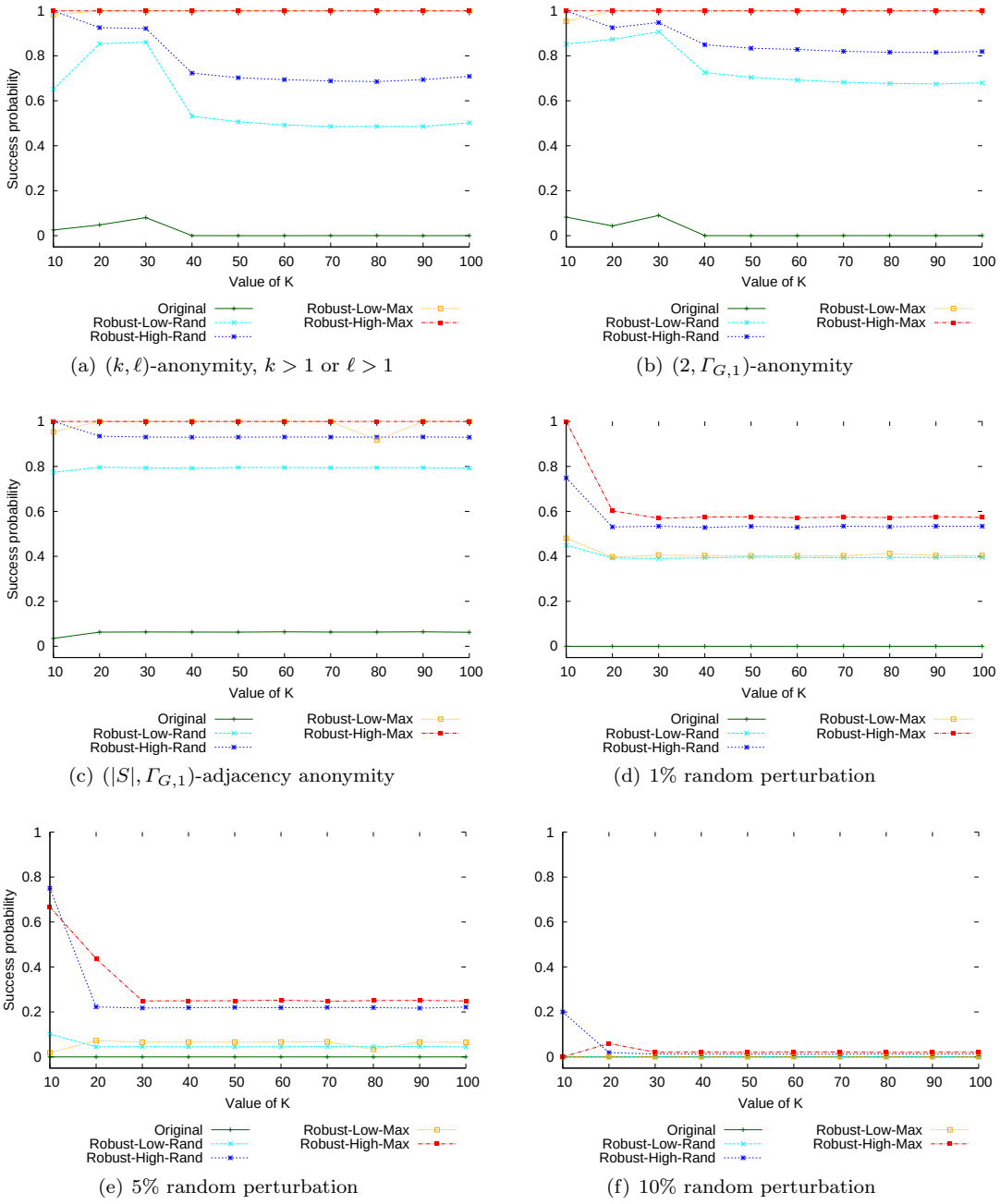
From the analysis of all results, an important and consistently occurring first observation is that the robust attack, in all its variants, displays a larger success probability than the original walk-based attack. In fact, for many parameter settings in ER and WS synthetic graphs, the difference is extreme, as the best-performing robust attack (Robust-High-Max) displays an average success probability close to 1, whereas the original attack displays an average success probability close

to 0. Another important observation that holds in all collections of synthetic graphs (see Figures 5 to 9, items (d) and (e)) is that even 1% of random noise completely thwarts the original attack, whereas different variants of the robust attack still perform at around 0.4 to 0.6 average success probabilities. In fact, the attack variants Robust-High-Max and Robust-High-Rand, that is the ones featuring large tolerance thresholds, still perform acceptably well on all synthetic graphs with a 5% random perturbation. An extreme case of resilience in the presence of noise is observed on real-life networks. Because of the large sizes of these graphs, the percentages of noise injected in these experiments translate into an enormous amount of modifications, and even in this case the robust attack variants manage to be successful in a small, but non-zero, fraction of cases in the URV graph, whereas the original attack is again completely thwarted. Also note that, in the case of the anonymisation methods (a) to (c), which also perform a considerably large number of perturbations on the real-life graphs, all variants of the robust attack display success probabilities ranging from medium to high, and continue to largely outperform the original walk-based attack.

Summing up the findings discussed in the previous paragraph, on the one hand we corroborate the previously reported fact that the original walk-based attack shows low resilience against the addition of even small amounts of random noise. On the other hand, and more importantly, the results obtained here support our claim that low resilience is not an inherent property of the active attack strategy itself, and that it is possible to design active attacks which are considerably more robust and thus represent a more serious threat in the context of noise addition approaches to privacy-preserving graph publication.

We now focus on the suitability of robust active attacks as a more appropriate benchmark, in comparison with the original attack, for evaluating anonymisation methods based on formal privacy guarantees. By analysing the results in items (a), (b) and (c) of Figures 5 to 8 and items (b) and (c) of Figure 9, we can see that the anonymisation methods were almost fully ineffective against the best performing robust attack (Robust-High-Max in most cases and Robust-High-Rand in the remaining cases). These results are consistent with the formal privacy guarantees provided by the anonymisation algorithms, since in all cases they only guarantee full protection from an attacker leveraging one sybil node, whereas all attacks displayed in these figures were conducted with 8 sybil nodes. On the other hand, the original walk-based attack is easily thwarted by most instances of all anonymisation algorithms. This behaviour of the original attack had already been reported (Mauw et al., 2016, 2018b), and its causes discussed. As the authors of these studies explain, this better-than-expected performance of the anonymisation methods was a side effect of the disruptions they caused in the graph, rather than a consequence of the formal privacy guarantees themselves. In other words, the low resilience of the attack played a more important role in the apparent effectiveness of the anonymisation methods than their formal privacy guarantees. Another undesirable effect of this problem is that it may enable misleading conclusions in comparing formally equivalent algorithms. For example, Mauw et al.'s algorithms for obtaining a (k, ℓ)-anonymous graph with $k > 1$ or $\ell > 1$ (Mauw et al., 2016) and a $(2, \Gamma_{G,1})$-anonymous graph (Mauw et al., 2018b) provide equivalent formal privacy guarantees. However, as seen in items (a) and (b) of Figure 7 and, to a lesser extent, in items (a) and (b) of Figure 5, there are some graph families where one of the two algorithms performs better than the other in terms of resistance to the original walk-based attack. These differences are not a consequence of the privacy guarantees provided by the algorithms. Instead, they are a consequence of the number of modifications that each method performs. As can be observed in the figures, the performance of both algorithms in terms of resistance to the robust attack Robust-High-Max is the same, which is consistent with the fact that both methods provide equivalent formal privacy guarantees. In scenarios like the one described here, researchers studying both methods would benefit from using the attack Robust-High-Max as comparison benchmark, since that would largely attenuate potentially misguiding side effects.

Next, we will discuss the effectiveness of the robust active attack variants in different types of graphs. From the analysis of Figures 5 to 8, items (a), (b) and (c), we can see that, on anonymised Erdős-Rényi and Watts-Strogatz graphs, the tolerance threshold plays a discrete role in differentiating the attacks leveraging maximally separated fingerprints. On the contrary, a high tolerance threshold does improve the effectiveness of the attack leveraging randomly generated



(a) (k, ℓ)-anonymity, $k > 1$ or $\ell > 1$

(b) $(2, \Gamma_{G,1})$-anonymity

(c) $(|S|, \Gamma_{G,1})$-adjacency anonymity

(d) 1% random perturbation

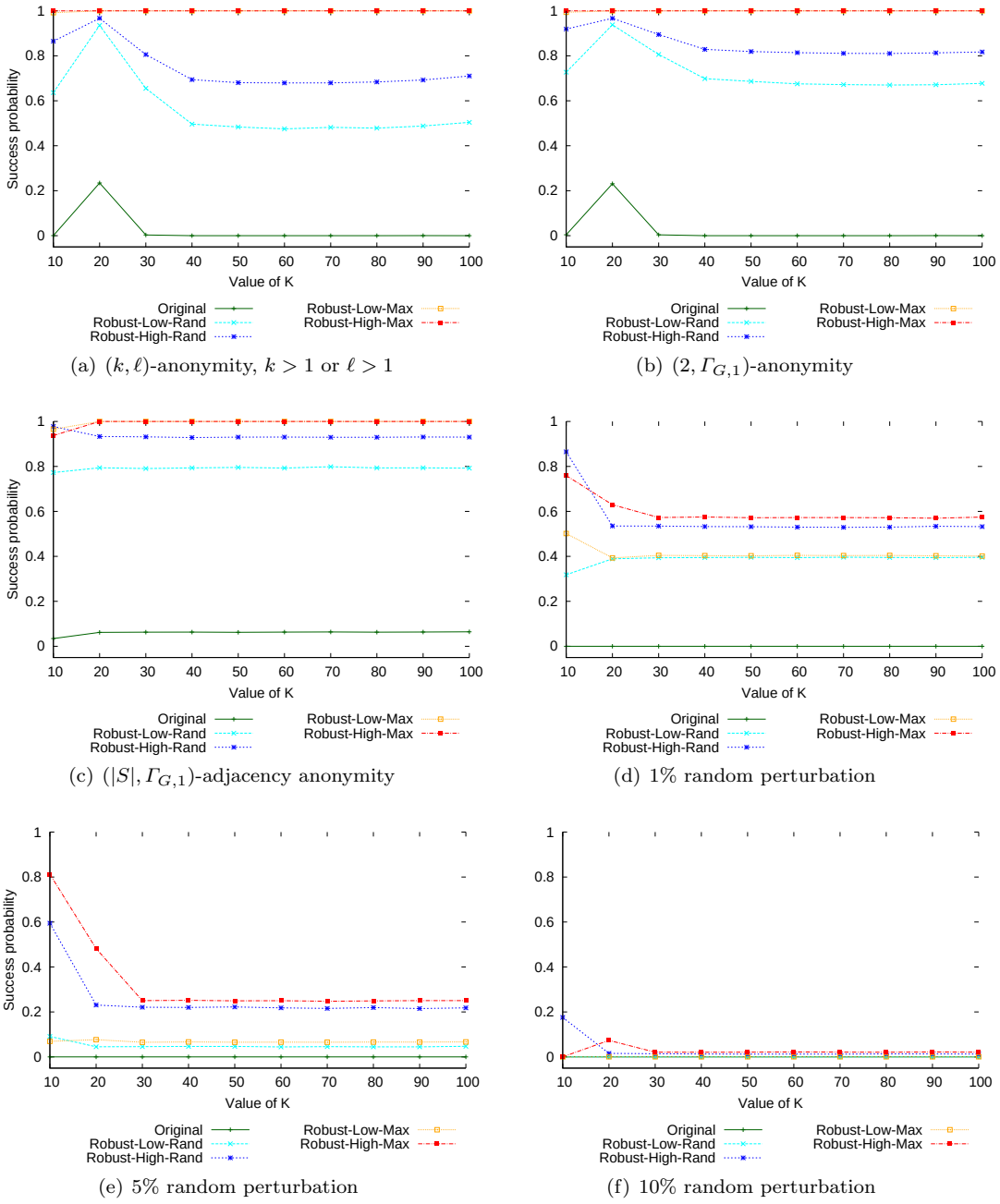
(e) 5% random perturbation

(f) 10% random perturbation

Fig. 7 Success probabilities of every attack variant on the collection of Watts-Strogatz small-world random graphs, with $\varrho = 0.5$, after publishing the graphs perturbed by the methods listed above.

fingerprints. Furthermore, in the case of randomly perturbed synthetic graphs of all types (see Figures 5 to 9, items (d) and (e)), the choice of a high tolerance threshold does lead to larger success probabilities in all cases. These observations highlight the central role of noise tolerance as a resilience-improving factor in robust attacks, especially when dealing with perturbation techniques based on noise addition.

To conclude our analysis, we focus on the behaviour of all robust attack variants on two scenarios: Barabási-Albert synthetic graphs anonymised with Mauw et al.'s method for enforcing $(k > 1, \ell > 1)$-anonymity (Mauw et al., 2016) (see Figure 9 (a)), and the Panzarasa and URV



(a) (k, ℓ)-anonymity, $k > 1$ or $\ell > 1$

(b) $(2, \Gamma_{G,1})$-anonymity

(c) $(|S|, \Gamma_{G,1})$-adjacency anonymity

(d) 1% random perturbation

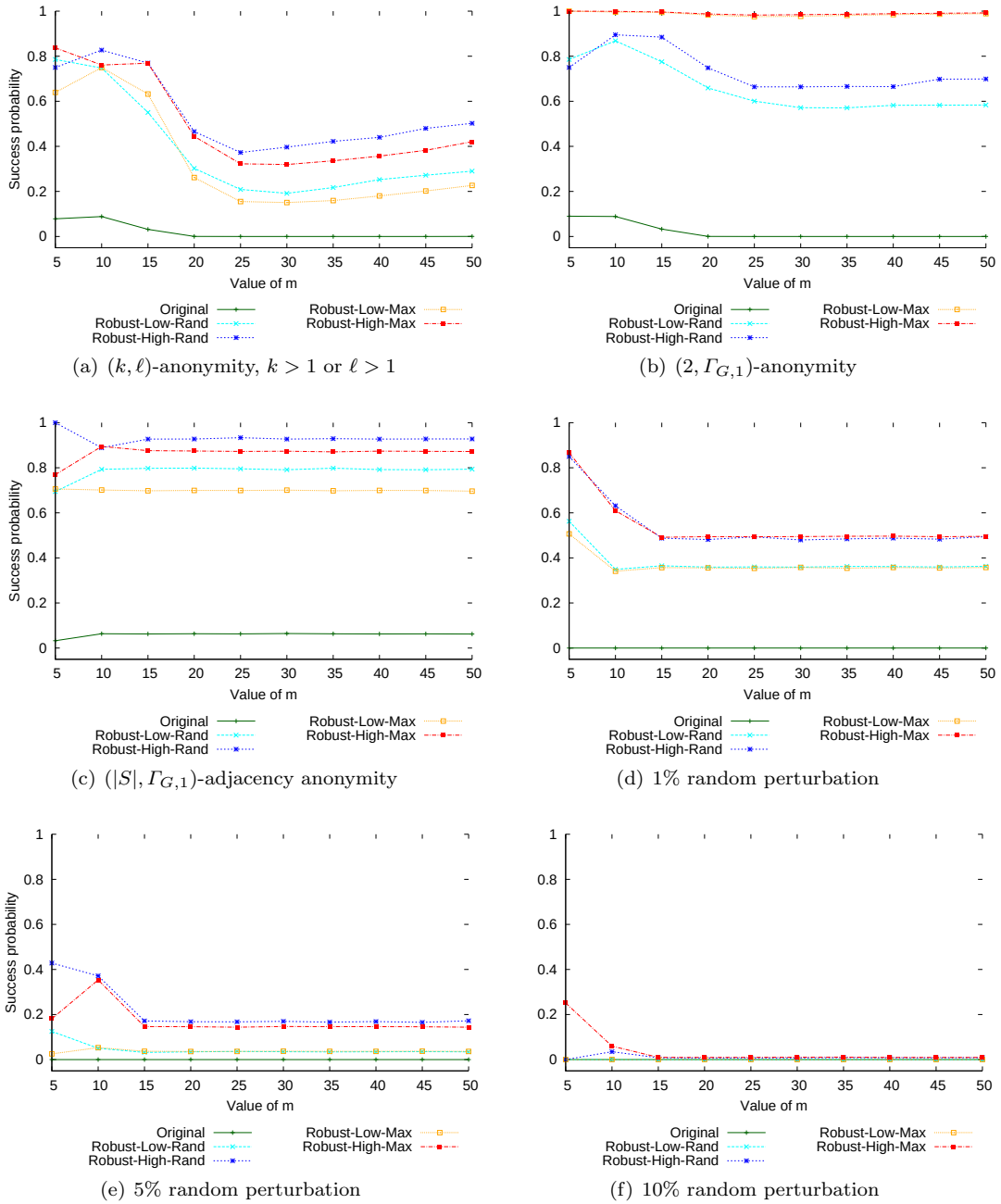
(e) 5% random perturbation

(f) 10% random perturbation

Fig. 8 Success probabilities of every attack variant on the collection of Watts-Strogatz small-world random graphs, with $\varrho = 0.75$, after publishing the graphs perturbed by the methods listed above.

graphs anonymised with Mauw et al.'s method for enforcing $(k = |S|, \Gamma_{G,1})$-anonymity (Mauw et al., 2018b) (see Table 2). These two scenarios are the only cases where all variants of the robust attack are collectively thwarted to a reasonable extent by an anonymisation method based on a formal privacy guarantee. The relevance of these observations lies in the hints that they provide for the design of new anonymisation algorithms to successfully counteract robust active attacks. Analysing the causes for the lower success probability of the robust attacks in these scenarios, we observed that in both cases the number of vertices which were structurally similar to some of the sybil nodes was larger. Moreover, this trend was reinforced by the anonymisation methods,

which resulted in the retrieval of a much larger number of false positives during the sybil subgraph retrieval stage of the attack. That is, even though the attacks succeeded in finding the correct sybil subgraph, the number of subgraphs at the same edit distance from the original sybil subgraph was larger, which made the probability of selecting the correct one smaller. Intuitively, this observation suggests that enforcing indistinguishability of potential sybil nodes in the perturbed graph may be a better countermeasure against robust active attacks than injecting noise, since even in the case that the attack retrieves the sybil subgraph, it will not be able to distinguish it from the (potentially numerous) equally similar false positives, thus reducing the success probability.



(a) (k, ℓ)-anonymity, $k > 1$ or $\ell > 1$

(b) $(2, \Gamma_{G,1})$-anonymity

(c) $(|S|, \Gamma_{G,1})$-adjacency anonymity

(d) 1% random perturbation

(e) 5% random perturbation

(f) 10% random perturbation

Fig. 9 Success probabilities of every attack variant on the collection of Barabasi-Albert scale-free random graphs, after publishing the graphs perturbed by the methods listed above.

Perturbed network		Original	Robust-Low-Rand	Robust-High-Rand	Robust-Low-Max	Robust-High-Max
Panzarasa	(a)	0.4272	0.9087	0.9259	0.9500	0.9423
	(b)	0.6393	0.9627	0.9589	0.9467	0.9563
	(c)	0.0082	0.4900	0.5000	0.5722	0.7500
	(d)	0.0000	0.0000	0.0000	0.0000	0.0000
	(e)	0.0000	0.0000	0.0000	0.0000	0.0000
	(f)	0.0000	0.0000	0.0000	0.0000	0.0000
URV	(a)	0.4587	0.9308	0.9248	0.9326	0.9368
	(b)	0.6039	0.9534	0.9522	0.9280	0.9335
	(c)	0.0081	0.4046	0.4524	0.4730	0.4500
	(d)	0.0000	0.0001	0.0056	0.0001	0.0061
	(e)	0.0000	0.0000	0.0000	0.0000	0.0000
	(f)	0.0000	0.0000	0.0000	0.0000	0.0000

Table 2 Success probabilities of every attack variant on real-life social networks.

7 Conclusions

In this study, we have re-assessed the capabilities of active attackers in the setting of privacy-preserving publication of social graphs. In particular, we have given definitions of robustness for different stages of the active attack strategy and have shown, both theoretically and empirically, scenarios under which these notions of robustness lead to considerably more successful attacks. One particular criticism found in the literature, that of active attacks lacking resilience even to a small number of changes in the network, has been shown in this paper not to be an inherent problem of the active attack strategy itself, but rather of specific instances of it. In light of the results presented here, we argue that active attacks should receive more attention by the privacy-preserving social graph publication community. In particular, existing privacy properties and anonymisation algorithms designed to counteract active attacks should be revised, and new ones should be devised, to account for the capabilities of robust active attackers.

Acknowledgements: We thank the anonymous reviewers for their valuable comments and suggestions. The work reported in this paper received funding from Luxembourg's Fonds National de la Recherche (FNR), via grant C17/IS/11685812 (PrivDA).

References

Backstrom L, Dwork C, Kleinberg J (2007) Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: Procs. of the 16th Int'l Conf. on World Wide Web, New York, NY, USA, pp 181–190, DOI 10.1145/1242572.1242598

Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512

Bunke H (2000) Recent developments in graph matching. In: Procs. of the 15th Int'l Conf. on Pattern Recognition, pp 117–124

Casas-Roma J, Herrera-Joancomartí J, Torra V (2013) An algorithm for k-degree anonymity on large networks. In: Procs. of the 2013 IEEE/ACM Int'l Conf. on Advances in Social Networks Analysis and Mining, pp 671–675

Casas-Roma J, Herrera-Joancomartí J, Torra V (2017) k-degree anonymity and edge selection: improving data utility in large networks. Knowledge and Information Systems 50(2):447–474

Cheng J, Fu AWc, Liu J (2010) K-isomorphism: privacy preserving network publication against structural attacks. In: Procs. of the 2010 ACM SIGMOD Int'l Conf. on Management of Data, pp 459–470

Chester S, Kapron BM, Ramesh G, Srivastava G, Thomo A, Venkatesh S (2013) Why waldo befriended the dummy? k-anonymization of social networks with pseudo-nodes. Social Network Analysis and Mining 3(3):381–399

Collberg C, Kobourov S, Carter E, Thomborson C (2003) Error-correcting graphs for software watermarking. In: Procs. of the 29th Workshop on Graph Theoretic Concepts in Computer Science, pp 156–167

Eppstein D, Goodrich MT, Lam J, Mamano N, Mitzenmacher M, Torres M (2016) Models and algorithms for graph watermarking. In: Procs. of the Int'l Conf. on Information Security, pp 283–301

Erdős P, Rényi A (1959) On random graphs. Publicationes Mathematicae Debrecen 6:290–297

Fober T, Klebe G, Hüllermeier E (2013) Local clique merging: An extension of the maximum common subgraph measure with applications in structural bioinformatics. In: Algorithms from and for Nature and Life, Springer, pp 279–286

Guimera R, Danon L, Diaz-Guilera A, Giralt F, Arenas A (2003) Self-similar community structure in a network of human interactions. Physical review E 68(6):065103

Hay M, Miklau G, Jensen D, Towsley D, Weis P (2008) Resisting structural re-identification in anonymized social networks. Proceedings of the VLDB Endowment 1(1):102–114, DOI 10.14778/1453856.1453873

Ji S, Li W, Mittal P, Hu X, Beyah RA (2015) Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization. In: Procs. of the 24th USENIX Security Symposium, pp 303–318

Jorgensen Z, Yu T, Cormode G (2016) Publishing attributed social graphs with formal privacy guarantees. In: Procs. of the 2016 Int'l Conf. on Management of Data, pp 107–122

Karwa V, Slavković AB (2012) Differentially private graphical degree sequences and synthetic graphs. In: Procs. of the Int'l Conf. on Privacy in Statistical Databases, pp 273–285

Liu C, Mittal P (2016) Linkmirage: Enabling privacy-preserving analytics on social relationships. In: Procs. of the Network and Distributed System Security Symposium, DOI 10.14722/ndss.2016.23277

Liu K, Terzi E (2008) Towards identity anonymization on graphs. In: Procs. of the 2008 ACM SIGMOD Int'l Conf. on Management of Data, New York, NY, USA, pp 93–106, DOI 10.1145/1376616.1376629

Lu X, Song Y, Bressan S (2012) Fast identity anonymization on graphs. In: Procs. of the Int'l Conf. on Database and Expert Systems Applications, pp 281–295

Ma T, Zhang Y, Cao J, Shen J, Tang M, Tian Y, Al-Dhelaan A, Al-Rodhaan M (2015) Kdvem: a k-degree anonymity with vertex and edge modification algorithm. Computing 97(12):1165–1184

Mallek S, Boukhris I, Elouedi Z (2015) Community detection for graph-based similarity: application to protein binding pockets classification. Pattern Recognition Letters 62:49–54

Mauw S, Trujillo-Rasua R, Xuan B (2016) Counteracting active attacks in social network graphs. In: Procs. of the 30th Annual IFIP WG 11.3 Conf. on Data and Applications Security and Privacy, Lecture Notes in Computer Science, vol 9766, pp 233–248

Mauw S, Ramírez-Cruz Y, Trujillo-Rasua R (2018a) Anonymising social graphs in the presence of active attackers. Transactions on Data Privacy 11(2):169–198

Mauw S, Ramírez-Cruz Y, Trujillo-Rasua R (2018b) Conditional adjacency anonymity in social graphs under active attacks. Knowledge and Information Systems (in press), DOI 10.1007/s10115-018-1283-x

Mittal P, Papamanthou C, Song D (2013) Preserving link privacy in social network based systems. In: Procs. of the Network and Distributed System Security Symposium

Narayanan A, Shmatikov V (2009) De-anonymizing social networks. In: Procs. of the 30th IEEE Symposium on Security and Privacy, pp 173–187, DOI 10.1109/SP.2009.22

Panzarasa P, Opsahl T, Carley KM (2009) 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, DOI 10.1002/asi.v60:5

Peng W, Li F, Zou X, Wu J (2012) Seed and grow: An attack against anonymized social networks. In: Procs. of the 9th Annual IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc

Communications and Networks, pp 587–595
Peng W, Li F, Zou X, Wu J (2014) A two-stage deanonymization attack against anonymized social networks. IEEE Transactions on Computers 63(2):290–303, DOI 10.1109/TC.2012.202
Rousseau F, Casas-Roma J, Vazirgiannis M (2017) Community-preserving anonymization of graphs. Knowledge and Information Systems 54(2):315–343
Sala A, Zhao X, Wilson C, Zheng H, Zhao BY (2011) Sharing graphs using differentially private graph models. In: Procs. of the 2011 ACM SIGCOMM Conf. on Internet Measurement, pp 81–98
Salas J, Torra V (2015) Graphic sequences, distances and k-degree anonymity. Discrete Applied Mathematics 188:25–31
Samarati P (2001) Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering 13(6):1010–1027, DOI 10.1109/69.971193
Sanfeliu A, Fu K (1983) A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, Man, and Cybernetics 13(3):353–362, DOI 10.1109/TSMC.1983.6313167
Sweeney L (2002) k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5):557–570
Trujillo-Rasua R, Yero IG (2016) k-metric antidimension: A privacy measure for social graphs. Information Sciences 328:403–417, DOI 10.1016/j.ins.2015.08.048
Varrette S, Bouvry P, Cartiaux H, Georgatos F (2014) Management of an academic HPC cluster: The UL experience. In: Procs. of the 2014 Int'l Conf. on High Performance Computing & Simulation, Bologna, Italy, pp 959–967
Wang Y, Xie L, Zheng B, Lee KC (2014) High utility k-anonymization for social network publishing. Knowledge and Information Systems 41(3):697–725
Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440
Xue M, Karras P, Raïssi C, Kalnis P, Pung HK (2012) Delineating social network data anonymization via random edge perturbation. In: Procs. of the 21st ACM Int'l Conf. on Information and Knowledge Management, pp 475–484, DOI 10.1145/2396761.2396823
Yu H, Kaminsky M, Gibbons PB, Flaxman A (2006) Sybilguard: defending against sybil attacks via social networks. In: Procs. of the 2006 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, Pisa, Italy, pp 267–278
Yu H, Gibbons PB, Kaminsky M, Xiao F (2008) Sybillimit: A near-optimal social network defense against sybil attacks. In: Procs. of the 2008 IEEE Symposium on Security and Privacy, Oakland, CA, USA, pp 3–17
Zhang J, Cormode G, Procopiuc CM, Srivastava D, Xiao X (2015) Private release of graph statistics using ladder functions. In: Procs. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data, pp 731–745
Zhao X, Liu Q, Zheng H, Zhao BY (2015) Towards graph watermarks. In: Procs. of the 2015 ACM Conf. on Online Social Networks, pp 101–112
Zhou B, Pei J (2008) Preserving privacy in social networks against neighborhood attacks. In: Procs. of the 2008 IEEE 24th Int'l Conf. on Data Engineering, Washington, DC, USA, pp 506–515, DOI 10.1109/ICDE.2008.4497459
Zou L, Chen L, Özsu MT (2009) K-automorphism: A general framework for privacy preserving network publication. Proceedings of the VLDB Endowment 2(1):946–957, DOI 10.14778/1687627.1687734