"International Journal of Information Security" manuscript No.
(will be inserted by the editor)

Hiding Information Against Structural Re-identification

Gábor György Gulyás · Sándor Imre

Received: 2015-10-07 / Accepted: 2018-02-01

Abstract Connections between users of social networking services pose a significant privacy threat. Recently, several social network de-anonymization attacks have been proposed that can efficiently re-identify users at large-scale, solely considering the graph structure. In this paper, we consider these privacy threats, and analyze de-anonymization attacks at the model level against a user-controlled privacy-enhancing technique called identity separation. The latter allows creating seemingly unrelated identities in parallel, even without the consent of the service provider or other users.

It has been show that identity separation can be used efficiently against re-identification attacks if user cooperate with each other. However, while participation would be crucial, this cannot be granted in a real life scenario. Therefore, we introduce the y-identity model, in which the user creates multiple separated identities, and assigns the sensitive attribute to one of them according to a given strategy. For this, we propose a strategy to be used in real life situations, and formally prove that there is a higher bound for the expected privacy loss which is sufficiently low.

Keywords anonymity · privacy · social networks · re-identification · identity separation

G. Gy. Gulyás
Privatics Team, INRIA
E-mail: gabor.gulyas@inria.fr

S. Imre
Mobile Communications and Quantum Technologies Laboratory (MCL), Dept. of Networked Systems and Services, BME
E-mail: imre@hit.bme.hu

1 Introduction

Social media services are used every day by millions. However, besides the added value these services provide, social media also serves as an optimal platform for commercial surveillance, and as recent cases show, for government surveillance [33]. Forms of commercial surveillance may be implicitly provided by the social networking service provider; e.g., business partners or scientific collaborators can be allowed to access sanitized data chunks from time to time, which may be abused later. Social meta-data can additionally be put into the use of re-identification of individuals in such anonymized datasets [30] or even in datasets of mobility traces [8, 23, 38].

Naive data anonymization techniques cannot provide an acceptable level of protection. Several works have proven that nodes in anonymized datasets (also called as sanitizated datasets) can be re-identified with high accuracy in various contexts [3, 4, 9, 13, 22, 23, 29, 30, 34, 35, 38]. Most of these methods are capable of achieving large-scale re-identification of social datasets consisting of hundred thousand records.

In particular, we consider de-anonymization attacks that use structural information only for re-identification of anonymous entities within large datasets [3, 23, 29, 30, 34, 35, 38]. There are recent examples, when the use of these algorithms is extended to services containing meta-data reflecting the underlying social connections between its entities can be targeted. For example, it has been shown, that spatio-temporal datasets (like mobility traces or check-ins) can be converted into a social network graph [36], which can be then aligned with another social network in order to recover identities [23, 30].

We demonstrate the underlying principles of social network re-identification on the following example. Let us consider an attacker who buys an anonymous social graph containing political preferences (as in Fig. 1b). While this dataset can be useful for analysis alone itself, it would be even more valuable if assigning each node to a public identity would be possible. After crawling social relations from another source, for instance from a publicly available online social networking site, the re-identification process can be done in two steps.

First, the attacker can search for nodes with outstanding properties, like using node degree (degree of a node denotes the number of contacts it has). The attacker can create a re-identification match between the nodes $v_{Dave} \leftrightarrow v_3$ and $v_{Fred} \leftrightarrow v_2$ as they are high degree nodes who are unique in the networks. However, this cannot be continued further. For example, v_{Harry} has two connections, but so does v_{Carol}. However, if we consider that v_{Harry} is connected to both v_{Dave}, v_{Fred}, this boils down our choices to the re-identification mapping of $v_{Harry} \leftrightarrow v_1$.

There are several ways to combat these kinds of attacks. In this paper, we focus on a user centered technique, called identity separation, that could be applied to existing services even without modification of the service itself. Identity separation can be done on the client side, it could be used without getting the consent of the service provider. Identity separation is based on the concept how we use our real identities in everyday life: we share different information in different situations and with different acquaintances [11]. This can also be applied to social networks to segregate information with different groups of contacts [20]. In our previous works we have proposed models for applying identity separation to social networks [14, 20], and also provided the analysis of identity separation against re-identification at a model level [16, 18]. In these evaluations users adopted identity solely on their own, and in some of the settings they cooperated to stop the attack.

Identity separation is not solely a technical innovation: it already exists and it is in use in real-world scenarios. There is a long list of authors who used pen names for several reasons[1], e.g., to protect their original identity, or used multiple pen names to avoid harming the reputation of each identity. Identity separation still has its uses today, let us just think of the separation of business and private identities [21] (e.g., via Facebook and LinkedIn). It can be useful also when it is suspected that two businesses exchange data of their users. Such an exchange could cause economic disadvantages for the users, thus using different account names, emails can be considered beneficial (e.g., using solutions such as Albine's Maskme[2]).

In this paper, we provide the model level analysis of identity separation as a privacy-enhancing tool against de-anonymization attacks. We build on our previous work, as we use the identity separation models from [14]. Our aim is to provide guarantees compared to previous models of identity separation [16, 18]. We analyze identity separation from an individual point of view, where the goal is to minimize possible private data leakage. Our aim is to provide strategic guidelines for managing private information in partial identities for users who act alone.

Our main contributions are the following. We seek strategies for users acting on their own, and first we apply the underlying principle of k-anonymity, a common technique used for anonymization [40] to the structural re-identification context: the user creates an identity for which there are $k-1$ structurally indistinguishable other users. We find that this method is inadequate in the current context due to the diverse structure of networks. Therefore, we propose an alternative privacy-enhancing method, the novel information hiding technique called y-identity, in which the user creates y partial identities in parallel and hides a sensitive information in one of them. We analyze this technique for different types of attackers, and propose a strategy for unknown attackers. We prove that by using this strategy, the expected privacy loss is equal or lower proportionally to $\frac{1}{y}$, which is at least equal, but likely better than what is provided by the k-anonymity model.

The paper is organized as follows. Section 2 discusses related work, and in Section 3 we present the threat model, notations, methodology we used in our work. In Section 4 and 5 we provide the analysis on the k-anonymity and the y-identity models. We discuss how the presented results could be applied in Section 6, and conclude our work in Section 7.

2 Related Work

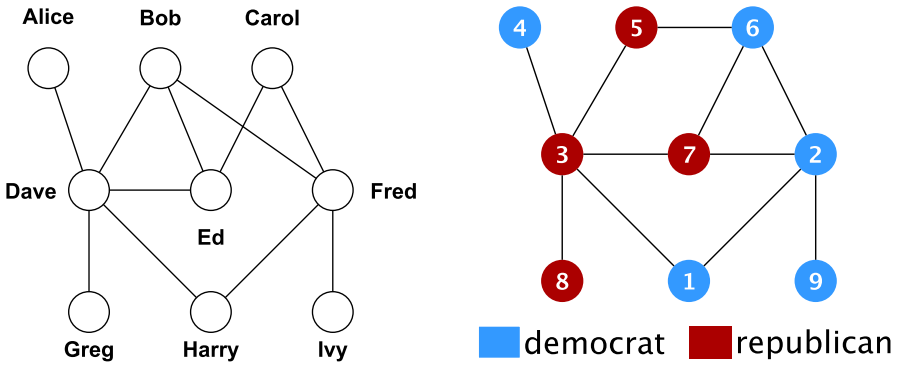
2.1 De-anonymization Attacks

In the context of our paper, re-identification is the method for revealing the real identities of nodes within an anonymized graph (the sanitized target graph) by using a social network obtained from an auxiliary source (the source graph, also called background knowledge).

In their original experiment the authors of [30] used 4-cliques of high degree nodes to initialize their attack,

[1] Wikipedia on pen names: http://en.wikipedia.org/wiki/Pen_name

[2] Albine Maskme providing disposable emails: https://www.abine.com/maskme/



(a) Social network crawled from public source (auxiliary data)

(b) Anonymous data published with political preferences

Fig. 1: For example, an attacker can buy anonymized social network data for analysis of political preferences (b). Next, using structural information obtained from a public social networking site (a) he can try to re-identify nodes with political preferences.

to which we later refer to as Nar09. The initial re-identification is followed by a sequence of propagation steps (propagation phase). These steps are iterated on the neighbors of the nodes already re-identified until new nodes can be re-identified (i.e., it continuously extends the seed set). Identified nodes are also revisited. In each iteration, candidates are selected from target graph nodes, which share at least a common mapped neighbor with the source node being re-identified. Target candidates are then compared by scoring their similarity to the source node. If there is an outstanding candidate, the source and target graphs are changed, and a reverse checking is executed in order to verify the proposed mapping. If the result of reverse checking equals the source node, this is accepted as a valid mapping. As the seeding identifies globally outstanding nodes, and the propagation examines nodes connecting to the ones already re-identified (i.e., locally outstanding ones), these phases can also be named as global and local re-identification phases [15].

It has been shown by Srivatsa and Hicks that location traces can also be re-identified with similar methods [38]. In their work on small datasets (125 nodes and below), they succeeded in identifying circa. 80% of users by building anonymous networks of location traces, and using explicit social networks for de-anonymization. The work of Pham et al. showed that these algorithms using spatio-temporal data for making social network connections, can be extended to larger datasets [36]. Building upon their work, Ji et al. showed that spatio-temporal data at the scale of hundred thousand entities can be easily re-identified [23]. In their work, first a social network is generated based on the inspection of co-occurrences in the spatio-temporal dataset, then it is re-identified by using a social network as auxiliary data. However, these algorithms of [25, 26, 38] can also be used to attack regular social networks.

There are other works continuing the line of classical structural social de-anonymization algorithms. Narayanan et al. in 2011 presented another variant of their attack [29] specialized for the task of working on two snapshots of the same network (with a higher recall rate). Pedarsani et al. proposed a novel type of attack that can work without any initial input such as seeds [34]. The algorithm designed by Nilizadeh et al. uses other attacks as a base algorithm, and it exploits the cluster-oriented structure of the networks: runs the base re-identification algorithm first on the cluster structure, then inside them [31]. In their evaluation they used the Nar algorithm. The works of Yartseva and Grossglauser [42], and the paper of Korula and Lattenzi [27] contain simplified de-anonymization attacks in order to enable formal analysis of the algorithms. In our previous work we proposed Grasshopper [7] (also referred to as Grh), a robust attack algorithm that works with very small error rates (typically less then 1%).

The first comparative evaluation of most structural de-anonymization attacks were first provided in [24]. Seven algorithms in [25–27, 30, 34, 38, 42] were selected based on generality, scalability and practicality, were compared regarding robustness of background knowledge and against anonymization. Ji et al found that some algorithms were the most prominent only conditionally; they concluded that none of the evaluated al-

gorithms could be considered as generally better than others.

As none of the discussed attacks is proved to be better in general than the original attack Nar09 [30], later in our experiment we worked with this one. Identity separation could be considered to protect privacy against Nar09 and even stronger attacks. We propose such a method that we call the y-identity model, which allows information hiding even against strong adversaries.

2.2 User Centered Privacy Protection Methods

There are several ways for tackling re-identification attacks. However, as companies can be legally obliged to share private user data [33], we prefer not relying on the service provider for sanitization. Otherwise, one might consider using revised service models, such as distributed social networks [12], which could be a substitute to currently existing services.

We seek user centered privacy protection mechanisms instead: ones that can be applied to existing services (instead of graph sanitization applied by the service provider), ones that either hide user information or that are capable of preventing large-scale re-identification. For instance, Scramble is a good example for such solutions: it is independent of the service provider and allows a fine-grained access control for managing the sharing process of user data by encryption [6].

In another work, Beato et al. proposed the friend-in-the-middle model, where proxy-like nodes act as mediators to hide connections, successfully tackling the attack when approx. 10% adopt the technique [5]. The viability of the FiM model is demonstrated on two snapshots of the Slashdot network (obtained from the SNAP collection [37]). Identity separation has additional features, for example hiding profile information beside making relationships private [11]. Thus identity separation allows an even finer-grained management of information, with less cooperation compared to the friend-in-the-middle model (which required the cooperation of three for hiding a single edge).

Previously, we have analytically showed that identity separation is an effective tool against clique based seeding mechanisms [14]. We furthermore analyzed the protective strength of identity separation against the propagation phase of Nar09 with simulation on datasets obtained from three different social networks [16, 18]. We have shown that it is possible to stop the re-identification attacks just by having 3-50% users adopting identity separation (the number of participants depends if users are cooperating or not), and it is possible to effectively hide information even for a few nodes.

However, these methods worked without guarantees and required cooperation in order to be efficient. Thus, we seek solutions that can provide privacy by guaranteeing uncertainty of the attacker. Thus we propose the y-identity model that allows minimization of disclosed information on the individual level: both against the state-of-the-art attack and even stronger re-identification attacks.

In our current work, we build upon the probability based identity separaiton models we previously introduced in [14], due to the lack of real-world data. These models capture identity separation as splitting a node, and assigning edges to the new nodes. The number of new identities is modeled with a random variable $Y = y$, without requiring an exact distribution. Four models are distinguished according to edge sorting capabilities, depending on whether it is allowed to delete (i.e., an edge becomes private) or to duplicate edges. We used two of these in our experiments. The basic model is simple and easy to work with, as it simply redistributes edges between the new identities (no edge deletion or duplication allowed). We also used the best model describing privacy oriented user behavior (no edge duplication, but deletion allowed).

3 Methodology

In our work, we denote the sanitized graph to be deanonymized as G_{tar}, and the auxiliary data source as G_{src} (where node identities are known). $\tilde{V}_{src} \subseteq V_{src}, \tilde{V}_{tar} \subseteq V_{tar}$ denote the set of nodes that mutually exist in both networks (i.e., overlapping nodes). Ground truth is represented by mapping $\mu_G : \tilde{V}_{src} \rightarrow \tilde{V}_{tar}$ denoting relationship between coexisting nodes, and $\lambda_G : \tilde{V}_{src} \rightrightarrows \tilde{V}_{tar}$ denote mappings between nodes in G_{src} and the sets of their separated identities in G_{tar}. Running a deterministic re-identification attack on (G_{src}, G_{tar}) initialized by seed set $\mu_0 : V_{src} \rightarrow V_{tar}$ (i.e., set of initially re-identified nodes) results in a re-identification mapping denoted as $\mu : V_{src} \rightarrow V_{tar}$. We denote the set of nodes adopting identity separation as $V_{ids} \subseteq V_{tar}$.

During our experiments we used multiple datasets with different characteristics in order to avoid biases caused by the structure. These were large networks where brute-force attacks are practically not feasible. For keeping our measurements realistic, datasets were obtained from real networks. We used the Slashdot network crawled in 2009 (82,168 nodes, 504,230 edges) and the Epinions network crawled in 2002 (75,879 nodes, 405,740 edges) from the SNAP collection [37]. Our third dataset is a subgraph exported from the LiveJournal

network crawled in 2010 (by the authors; consisting of 66,752 nodes, 619,512 edges).

We generated test data for evaluating re-identification attacks as follows (in Section 5.7). First derived a background knowledge (G_{src}) and a target graph (G_{tar}) from the source dataset, having the desired fraction of nodes and edges overlapping, and then modeled identity separation on a subset of nodes in the target graph. For creating G_{src}, G_{tar}, we used the perturbation strategy proposed by Narayanan and Shmatikov [30], which produces realistic test data. Their algorithm works as follows. First, it derives G_{src}, G_{tar} with the desired fraction of overlapping nodes (α_v) from the source dataset. Next it deletes edges independently from these copies to achieve an edge overlap α_e. We could easily calculate ground truth μ_G by knowing the original graph structure.

After running several measurements with different settings for α_v, α_e, we choose $\alpha_v = 0.5$, $\alpha_e = 0.75$ for our experiments. This setting is a good trade-off at which a significant level of uncertainty is present in the data (thus life-like), but it is still possible to identify a large ratio of the co-existing nodes. Next, we modeled identity separation on the target graph by uniformly sampling nodes (where $deg(v) \geq 2$). Then nodes are split and their edges are sorted according to the settings of the currently used identity separation model. By recording these operations, we can extend the ground truth mapping μ_G with λ_G.

We used the following settings for simulations. One of the most important parameter of Nar09 is Θ, which controls the ratio of true positives (recall rate) and false positives (error rate). The lower Θ is the less accurate mappings the algorithm will accept. We measured fairly low error rates even for small values of Θ in our earlier works [16, 17], therefore we have choosen to work with $\Theta = 0.01$. With respecting results in [17], we applied random seed selection of high degree nodes selected from the top 25% (denoted as `random.25`). Seed set size was a thousand nodes, as this proved to be robust in all networks [17], and in combination with `random.25` it simulated a stronger attacker. We used top degree seed nodes as the simulation of an even stronger attackers (denoted as `top`). Seed nodes always represent users who have not committed identity separation.

4 Evaluation of k-anonymity

Previous works on identity separation [16, 18] clarified that identity separation requires cooperation to efficiently stop attacks, and while it can provide privacy protection, that comes without guarantees. For example, a stronger attacker could find the proper identity regardless of user efforts that worked before.

Thus, we focus on techniques providing strong privacy guarantees The first technique we discuss is k-anonymity [40], a simple model that is able to provide a given level of privacy, limited by parameter k. In case of k-anonymity, the user aligns one of her identities to its neighbors for hiding the sensitive attribute it has. The second technique is based on the idea that the user creates a number of new identities and hides the sensitive information in one of those (analogously to the parameter k in k-anonymity). The latter is discussed in Section 5.

The definition of k-anonymity is based on the concept of quasi-identifiers, which are constructed from attributes of a data entity (e.g., user as a database row or attributes of a web browsing agent). Attributes of a quasi-identifier are not reckoned as explicit identifiers, but being used together can enable identification. For example, based on 1990 US Census data, Sweeney showed that 87% of the US population can be identified with the quasi-identifier of {`5-digit ZIP, gender, date of birth`} [39].

Definition 1 *k-anonymity.* A dataset is k-anonymous if for all entries there are at least k-1 other entries with the same quasi-identifiers [40].

Despite it has been shown that the concept of k-anonymity is inappropriate for anonymizing data with high dimensionality [1], it is applied and analyzed in many contexts even for the sanitization of social network structural data [2]. There are also known weaknesses of k-anonymity, for example regardless of anonymization the attacker can still learn information: the distribution of the sensitive attributes in the k-anonymous groups can significantly deviate from the global distribution. Subsequent models aim to patch such vulnerabilities, such as l-diversity and t-closeness [28].

In order to achieve k-anonymity, previous work proposed isomorphism-based methods. In particular, k-automorphism [43] and k-isomorphism [10] have been proposed to protect structural privacy against re-identification attacks: these methods anonymize the whole network, where for each node there are $k-1$ structurally equivalent other nodes. Both methods partition the network, then modify the structure in (and between) the partitions to k-anonymize these subsets.

While these methods meet the guarantees that we are looking for, they also need to have an overview and access to the whole dataset. This makes interaction with the service provider inevitable, which is not acceptable for user-centric identity separation techniques, where

we expect users to be able to act on their own. Therefore we propose and analyze a method that applies k-anonymity strictly on the individual level.

During the re-identification matching in the attacks, nodes are compared to their friends-of-friends, to the 2-hop neighborhood. Therefore, based on this, we can extend the concept of k-anonymity to identities (or users) regarding their 2-hop neighborhood, in order to preserve user privacy against large-scale re-identification attacks.

Definition 2 $(k,2)$-*anonymity.* A user $v_n \in G$ is $(k,2)$-anonymous if there are at least k-1 other users having exactly the same neighborhood, i.e.,

$$|\{v_i : v_i \in \{G.nbrs(V_n) \setminus v_n\}, V_i = V_n\}| \geq k,$$

where $V_j = G.nbrs(v_j)$.

In other words, this definition means that a node should be structurally equivalent to k-1 other nodes that are at a 2-hops distance from him. We need an algorithm for this; one that satisfies the definition of $(k,2)$-anonymity, and is constrained in its operations to the 2-hop neighborhood of the node.

Thus, we have constructed an algorithm (Alg. 1), called `K-AnonymizeNode`, for finding $(k,2)$-anonymous settings for users planning to apply identity separation. The algorithm assumed to know the network structure in a 2-hop distance. Beside parameter k the algorithm also takes an input of c that gives the desired neighborhood size of the new identity. Then the algorithm seeks if there are k two-hop neighbors that have exactly c common neighbors with the user. If there are no users to propose, the algorithm proposes new connections in order to meet the criteria of $(k,2)$-anonymity.

We measured the possibility of $(k,2)$-anonymity in the three networks on $1,000$ nodes randomly sampled from each (with $deg(v) \geq 30$) for $c \in \{3,5,10,20\}$. The results of our experiments are shown on Fig. 2. We selected results from Epinions dataset with $k = 2$ for explanation on Fig. 2a. While in almost half of the cases with $c = 2$ it was possible to achieve anonymity without adding edges, this was rather not possible for larger values of c. We observed similar results in other networks, and also when analyzing whether this property differ as the network size change (see Fig. 2b). For greater (and practicable) values of k achieving anonymity required adding even more edges (or it was impossible to reach).

Therefore, we concluded that $(k,2)$-anonymity is an inappropriate option for individually protecting privacy, as the structure of social networks is not making such techniques feasible. Thus, we analyzed alternative methods providing an equivalent level of privacy: when the user creates several new identities of which some are fake.

5 Analysis of the y-identity Model

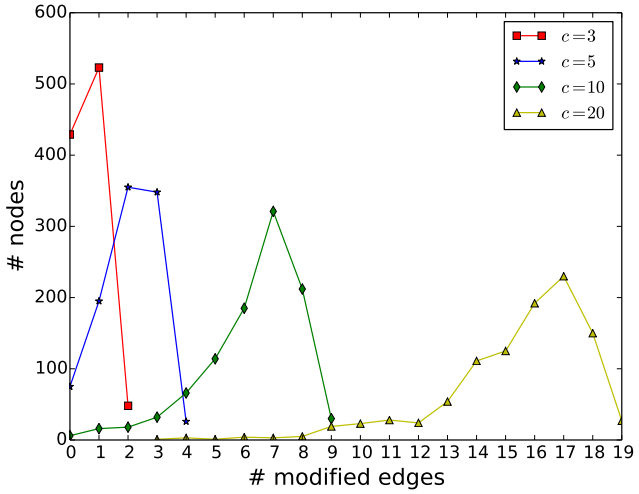
As k-anonymity failed, in the following we propose a novel method that provides privacy in a similar manner. In the new *y-identity model* the user creates y new identities and randomly assigns the privacy sensitive information to one of the identities (n.b. parameter y is used in a similar sense as k in the k-anonymity model is used: this parameter bounds the privacy the user can have). We assume the user is rational and aims to achieve optimal privacy-preserving settings. Thus, in our settings she would always choose a single identity for storing the sensitive value among all identities she possesses.

Several types of user data can be considered as a sensitive attribute: either sensitive personal attributes (e.g., religious or political preferences), free-text profile information (e.g., link to a website) or the content the user shares (e.g., wall messages). However, managing such a vast amount of information by hand can be difficult, and this process should be supported by an identity manager software (e.g., Scramble is such a proof-of-concept utility [6]). With such a support the user could be able to achieve fine-grained control over her profile and safely reveal the secret information only for the selected audience with ease, while separated identities would be represented as different users for the attacker, but also for the social network platform and its users.

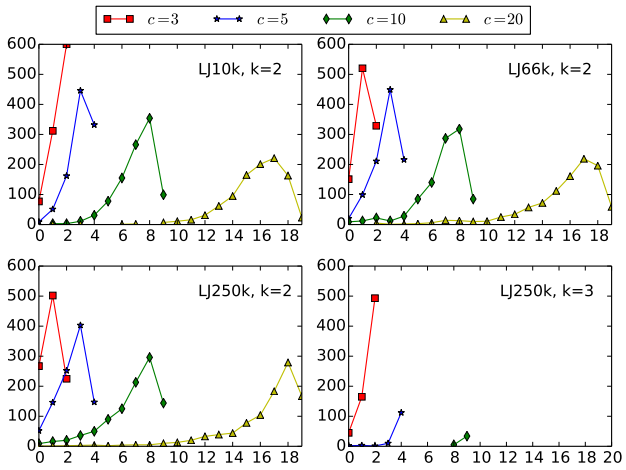
Definition 3 *y-identity.* A user is considered to be acting according to the y-identity model if she creates y separated identities (either in one or in multiple datasets), and assigns randomly a privacy-sensitive attribute to only one of the identities, determined by a given distribution. Other identities then receive plausible, but false attributes.

We consider mass re-identification attacks that aim to de-anonymize thousands of nodes in some sanitized networks [30]. A rational attacker aims to reveal quality private information at large in two sequential steps. First, the attacker uses a structural re-identification algorithm for discovering the mappings between the public identities of users and all of their separated identities in sanitized datasets. In Section 5.7 we provide a simple intuitive example for finding multiple separated identities by utilizing Nar09. Then, after finding these mappings for a given user, the attacker decides which identity has the correct sensitive attribute: she decides that either none, or she picks one of the partial identities to be valid.

In this paper, we consider the case when the attacker uses a regular social network as background knowledge. If the background knowledge would also have identity separated users, the attacker could find mappings be-



(a) Results from Epinions dataset with $k = 2$. While in almost half of the cases it was possible to achieve anonymity for new identities with a very small neighborhood ($c = 3$) without modification, this was rather not possible for larger values of c. As the desired size of the neighborhood grew, the number of edges to add also increased.



(b) These experiments indicate that the findings discussed related to (a) are also true for other networks even for different sizes. It is additionally shown that if we increase k the situation rapidly develop into an even worse scenario.

Fig. 2: $(k, 2)$-anonymity with edge modification in action. Results shows this method is not feasible due to the great diversity in network structure.

tween partial identities. However, this would not necessarily pose a privacy risk for the users: the attacker could only learn mappings between fake identities and anonymized partial identities and their sensitive attributes. Eventually, this would mean that the attacker could not learn new information by de-anonymization, as the sensitive attribute could not be linked to a globally recognized (or known) personal identity.

5.1 Formal Description of the Attack

Assume that the attacker had run a successful large-scale re-identification attack against a pair of networks. In the next step the goal is find the correct sensitive information for users that have multiple partial identities. Narrowing our focus down to a given user, we can formally describe this process as a game; however, we did not always model it as a game (see the attacker model in Section 5.2 for details). Therefore, we provide the formal description of this problem as for games, where the player set $\mathcal{P}$ contains the user and the attacker.

Before the attacker obtained the dataset, user v_n acted according to the y-identity model to protect her privacy. Initially, she creates y new identities, in a single service, or in multiple social network based services. These partial identities are denoted as $v_{n\setminus i}$ $(i \in [1, y])$. Then the sensitive information is randomly assigned to one of these identities with probability r_i. The selected identity is denoted as $v^{\star}_{n\setminus i}$. We model the user selection decision for hiding the sensitive attribute with $P(R = i) = r_i$, where we expected that $\sum_{\forall i} r_i = 1$ (n.b. this covers the case of a deterministic decision, too).

Since the attacker had run the re-identification algorithm already, she has to find $v^{\star}_{n\setminus i}$ among all $v_{n\setminus i}$ $(i \in [1, y])$. We assume that the attacker has no information about the identity separation process itself.

At this point, we model the attacker decisions with distribution $P(Q = i) = q_i$, the probability for accepting the sensitive attribute of re-identified partial identity $v_{n\setminus i}$ to be valid. We allow $\sum_{\forall i} q_i \leq 1$ for attackers, as they might not accept any of the found attributes to be valid. This could be because that all values are conflicting the background knowledge of the attacker.

The strategy set $\mathcal{S}$ covers selecting one of the identities the user has. From a user point of view this is for storing the sensitive attribute, while for an attacker this is for accepting a value as valid. In some cases the attacker only has access to $\mathcal{S}' \subset \mathcal{S}$, limiting the number of her possible decisions. We assume that the decisions are made in single round. The attacker could repeatedly make decisions in several rounds; however, as he cannot verify the currently accepted attribute, this would not contribute anything to the success of the attack.

Finally, we can introduce utility values (or payoffs) denoted as $\mathcal{U}$. Let denote u^{+}_{n} as the utility for the user in case of avoiding a privacy breach (false information is

learned by the attacker), and u_n^- for private information leakage. Similarly, we denote u_A^+ and u_A^- for the attacker learning valid or false information. The example of cases we consider is provided in Table 1 for $y = 2$ identities within a single dataset.

Payoffs can be quite asymmetric. For instance, a single node may not be very important for the attacker (as being only one of hundreds of thousands), while the targeted private value can be very important for the user. This can lead to asymmetry such as $u_A^- \ll u_n^-$.

5.2 Attacker Model

In our attacker model, we consider two types of attackers:

1. *Strong attackers*, who are able to discover all y identities of a given user v_n. The attacker knows he has access to all identities of v_n. As both the attacker and the user knows all the possible choices the other could make (or in other words both players know $\mathcal{S}$), we will use a game-theoretic approach to determine best strategies.
2. *Weak attackers*, who are able to reveal some of the identities (even perhaps all of them), but are uncertain if there are any additional undiscovered identities or not (e.g., as there might be further unknown datasets that the adversary is unaware of). More formally, while the user knows $\mathcal{S}$, the attacker only has access to $\mathcal{S}' \subseteq \mathcal{S}$, and does not know if $\mathcal{S}' = \mathcal{S}$. Due to missing possible pure strategies of the user and for the sake of simplicity, here we model the attacker as making decisions according to a given distribution on the discovered identities. For searching the best user strategy, we use an optimization approach for minimizing the expected privacy loss, where the user is assumed to be able to approximate the attacker's probabilistic decision function.

As a consequence, the y-identity model can provide two distinguished levels of privacy depending on the chosen adversary. If someone uses strategy against strong attackers that would allow a level of privacy against an attacker with no bounds on computation power and access to data. For these attackers, we assert that they always make a choice, i.e., $\sum_{\forall i} q_i = 1$.

For strategies against weak attackers, someone could have privacy against limited attackers, who cannot access all datasets or cannot re-identify all partial identities certainly. Furthermore, the decision making distribution of the weak attacker type could be estimated by the user by several means. For example, based on the background information they have (e.g., comparing the sensitive attributes to the background knowledge or global statistics of the network), by analyzing the validity of the information provided (e.g., consistency checking of sensitive attributes of all $v_{n\setminus i}$ with their neighborhood), or simply based on how the re-identification algorithm works (we give examples of this later).

As future work, it would be interesting to extend the attacker model with another type of weak attacker who can assess the probability that the sensitive information is stored in an identity that has not been found. Currently, this does not seem to be a reasonable assumption, however, this might be a subject to change. It can be also interesting to consider the re-identification algorithm as a part of the decision making process (instead of an initialization), and to see how the whole process could be analyzed as a game.

5.3 Evaluation of Strong Attackers

We model the evaluation of strong attackers as a single-round game between the attacker and the user ($\mathcal{P}$), where none of the players know the steps the other might have taken before. We call this the *identity partitioning game*, and it works as follows. The user assigns the sensitive information to the i^{th} partial identity $v_{n\setminus i}$ with probability r_i (changing the identity to marked as $v^{\star}_{n\setminus i}$). The attacker runs the re-identification attack that finds mappings to all partial identities, and will accept the sensitive attribute of one of them with probability q_i. Here the utility matrix is a diagonal matrix with the size of $(y \times y)$, having values as $(u_A^+; u_n^-)$ in the diagonal, and $(u_A^-; u_n^+)$ in all other places. Thus pure strategies $\mathcal{S}$ of the players, and utilities $\mathcal{U}$ are as discussed before.

The Nash equilibrium [32] of this game is a pair of strategies when none of the players can increase their payoff by modifying only their strategy alone. It can be easily concluded that no pure strategy equilibrium exists in this game; this would mean that both players would always select a fixed identity. If the user constantly chooses the i^{th} identity as her strategy, the attacker can respond by choosing the same identity, modifying the payoffs as $u_A^+; u_n^-$ favoring herself. Regardless how the user acts, the attacker could always have a response leading to an equivalent situation.

Fortunately, this could be easily fixed when the players make their decisions according to a certain distribution. In this case mixed strategy equilibrium exists with given optimal strategy probabilities.

Theorem 1 *A mixed strategy Nash equilibrium exists in the identity partitioning game (with a user having*

Table 1: Utility matrix ($\mathcal{U}$) for the case of $y = 2$.

		User	
		$v^\star_{n\setminus 1}$	$v^\star_{n\setminus 2}$
Attacker	$v^\star_{n\setminus 1}$	$u_A^+; u_n^-$	$u_A^-; u_n^+$
	$v^\star_{n\setminus 2}$	$u_A^-; u_n^+$	$u_A^+; u_n^-$

Table 2: The $\mathbf{q}_m$ vectors for the first example (Section 5.4.2) for all $\mathbf{m} = [m_1, m_2]$.

$\mathbf{m}$	0	1
0	$\mathbf{q}_{[0\ 0]} = [0\ 0]$	$\mathbf{q}_{[1\ 0]} = [q_1\ 0]$
1	$\mathbf{q}_{[0\ 1]} = [0\ q_2]$	$\mathbf{q}_{[1\ 1]} = [q_3\ q_4]$

Table 3: The $\mathbf{q}_m$ vectors for the second example (Section 5.4.3) for all $\mathbf{m} = [m_1, m_2]$.

$\mathbf{m}$	0	1
0	$\mathbf{q}_{[0\ 0]} = [0\ 0]$	$\mathbf{q}_{[1\ 0]} = [1\ 0]$
1	$\mathbf{q}_{[0\ 1]} = [0\ 1]$	$\mathbf{q}_{[1\ 1]} = [P_1\ P_2]$

y separated identities), where the equilibrium strategy probabilities are $q_i = \frac{1}{y}, r_i = \frac{1}{y}$ *(*$\forall i$*).*

Proof In order for the strategy of the user to be part of a Nash equilibrium, the expected payoff for each action of the attacker need to be indifferent. Comparing the expected payoff of the first strategy to all other strategies describes this criteria in the form of $y-1$ equations. These equations can be given as:

$$u_A^- r_i + \sum_{\forall k \neq i} u_A^+ r_k = u_A^- r_j + \sum_{\forall l \neq j} u_A^+ r_l, \tag{1}$$

where $i \neq j$. We can additionally use $\sum_{\forall i} r_i = 1$ as the y^{th} equation. Using the latter, prior equations in the form of Eq. (1) can be simplified as:

$$u_A^- r_i + u_A^+ (1 - r_i) = u_A^- r_j + u_A^+ (1 - r_j) \tag{2}$$

Using all of these equations, we have now a linear system of y equations, with the coefficient matrix is:

$$A = \left(\begin{array}{ccccccc|c} u_A^- - u_A^+ & u_A^+ - u_A^- & 0 & \cdots & 0 & & 0 & 0 \\ u_A^- - u_A^+ & 0 & u_A^+ - u_A^- & \cdots & 0 & & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots & \vdots \\ u_A^- - u_A^+ & 0 & 0 & \cdots & u_A^+ - u_A^- & & 0 & 0 \\ 1 & 1 & 1 & \cdots & 1 & & 1 & 1 \end{array}\right) \tag{3}$$

As all equations contribute a coefficient that is excluded from the others, we have a linear independent equation system. As we have y linearly independent equations and y variables, this system has a solution.

Equations in the form of Eq. (2) can be reduced to

$$r_i = r_j. \tag{4}$$

With $\sum_{\forall i} r_i = 1$ the only valid solution of the equation system is $r_i = \frac{1}{y}$ $(\forall i)$.

The equilibrium strategy can also be calculated identically for the attacker, due to the symmetry of the payoff matrix. Therefore, the Nash equilibrium strategy is at when both parties use a mixed strategy with probabilities $q_i = \frac{1}{y}, r_i = \frac{1}{y}$ $(\forall i)$. ■

Theorem 1 proves that the most efficient strategy one could find against strong attackers is to use random, equal assignment probabilities. While this is intuitively the best strategy, as we show later, it is not necessarily also the best for the weak type of attackers.

5.4 Evaluation of Weak Attackers

We evaluate weak attackers in the following. We assume that the user can estimate P_i, the discovery probabilities respectively of her partial identities $v_{n\setminus i}$ $(\forall i \in [1, y])$. Here, we work with P_i without restricting its value. However, the estimation of P_i depend mainly on two factors: the probability that the attacker can access the dataset that includes $v_{n\setminus i}$, and additionally the probability of finding that identity.

Later we show how lower estimates with Nar09 can be calculated for discovery probabilities within a single dataset. However, calculating P_i values precisely can be a hard task; in such a case, we propose to stick to the proposed solution we provide in Section 5.5.

5.4.1 Estimating the Expected Privacy Loss

Let start with a specific case when the attacker discovers some given identities of the user v_n, and calculate the estimated privacy loss for that situation. The fact of the discovery is stored in the discovery vector $\mathbf{m}$ (size of y), where $m_i \in \mathbf{m}$ represents whether the i^{th} identity ($v_{n\setminus i}$) was discovered or not ($m_i \in [0, 1]$, $m_i = 1$ indicating the identity was found). Then, the privacy loss depends if the sensitive information was put into one of the discovered identities, and the right one is accepted as valid.

Generally, the attacker decision can even vary depending which identities were discovered (i.e., based on $\mathbf{m}$). Therefore, we further refine the attacker decision distribution, and introduce the distribution vector $\mathbf{q_m}$, containing probabilities for a given instance of $\mathbf{m}$. For instance, the attacker may decide to choose uniformly between all discovered identities leading to different distributions depending on $\mathbf{m}$. Here $q_i^{\mathbf{m}} \in \mathbf{q_m}$ denotes the

probability that respecting $m_i \in \mathbf{m}$ the attacker accepts the sensitive information stored in $v_{n\setminus i}$ (n.b. $m_i = 0$ implies $q_i^{\mathbf{m}} = 0$).

The probability that the attacker obtains valid information in this case is $r_i \cdot q_i^{\mathbf{m}}$ for each discovered identity. Then we can describe the expected cost of privacy loss for a given $\mathbf{m}$ as:

$$u_n^- \cdot \left(\sum_{\forall i} r_i \cdot q_i^{\mathbf{m}} \cdot m_i \right), \forall i \in [1, y] \tag{5}$$

As $m_i = 0$ implies $q_i^{\mathbf{m}} = 0$, and otherwise $m_i = 1$, we leave m_i out from the formula in the following. The probability of having an instance of $\mathbf{m}$ can be described as follows:

$$P_{\mathbf{m}} = \prod_{\forall j} ((1 - m_j) + (-1)^{(1-m_j)} \cdot P_j), \forall j \in [1, y] \tag{6}$$

The expected privacy loss, iterating through the all available possibilities of $\mathbf{m}$ is as follows:

$$E_w[u_n] = \sum_{\forall \mathbf{m}} \left(\left(\prod_{\forall j} ((1 - m_j) + (-1)^{(1-m_j)} \cdot P_j) \right) \cdot \left(\sum_{\forall i} r_i \cdot q_i^{\mathbf{m}} \right) \right) \cdot u_n^- \tag{7}$$

where $i, j \in [1, y]$.

However, this formula leads to an interesting advise regarding the best user strategy: using pure strategies leads to less privacy loss against weak attackers.

Theorem 2 *Given a weak attacker with known $\boldsymbol{q_m}$ vectors (for all $\boldsymbol{m}$), a set of pure strategies $\mathcal{S}' \subseteq \mathcal{S}$ exists that should be used in order to minimize the expected privacy loss $E_w[u_n]$. Strategies in $\mathcal{S}'$ can be used either as pure strategies or as mixed strategies.*

Proof The formula in Eq. (7) can be rewritten in the following way:

$$\begin{aligned} E_w[u_n] &= u_n^- \cdot \sum_{\forall \mathbf{m}} \left(P_{\mathbf{m}} \cdot \sum_{\forall i} r_i \cdot q_i^{\mathbf{m}} \right) \\ &= u_n^- \cdot \sum_{\forall i} \left(\underbrace{\left(\sum_{\forall \mathbf{m}} q_i^{\mathbf{m}} \cdot P_{\mathbf{m}} \right)}_{\alpha_i} \cdot r_i \right) \end{aligned} \tag{8}$$

Term α_i is a known constant, thus in order to minimize privacy loss, we seek the minimum value of a linear sum with non-negative coefficients. The minimum depends on the user strategy probabilities. This value is minimal when:

$$\sum_{\forall j \in \arg\min_j \alpha_j} r_j = 1, \tag{9}$$

which means one of the following cases:

- If $|\arg\min_j \alpha_j| = 1$. Setting $r_j = 1$ where $j = \arg\min_j \alpha_j$, which is the equivalent of using a single pure strategy.
- If $|\arg\min_j \alpha_j| > 1$. Setting $\sum_{\forall j \in \arg\min_j \alpha_j} r_j = 1$, which is the equivalent either of using multiple specified strategies in an arbitrarily mixed way, or selecting one pure strategy from them.

■

Surprisingly, the conclusion of Theorem 2 contradicts to what we found for strong attackers: for weak attackers it is advised to use pure strategies instead of mixed ones. In specific cases, when there are multiple, equally good choices, mixed strategies can be based based on those strategies.

We note an interesting possible extension of this model. We could allow a third state with a positive probability r_0, when the sensitive information is not included any of the datasets. This extension would appear in decreasing all other r_i values ($\forall i > 0$) that appear in Eq. (7), implicitly decreasing the expected privacy loss value, too. Introducing such a state would mathematically suggest that users should maximize r_0, which is consistent with the common sense saying that if you want to have maximum privacy do not disclose sensitive content.

5.4.2 Example 1: Minimizing Cost in a Simple Case

In the following we provide a couple of examples demonstrating the use of the model, and we assume that the cost u_n^- does not differ for identities, and for keeping the calculations simple we use the cost uniformly as $u_n^- = 1$. In the first example we demonstrate the use of Eq. (7), with a single user having two identities ($y = 2$) in a single dataset.

For all combinations of $\mathbf{m}$, the $\mathbf{q}_m$ vectors can be defined as in Table 2. By using Table 2 and Eq. (7), the cost of privacy loss is characterized as:

$$\begin{aligned} E_w[u_n] =& P_1 \cdot (1 - P_2) \cdot r_1 \cdot q_1 + (1 - P_1) \cdot P_2 \cdot r_2 \cdot q_2 \\ &+ P_1 \cdot P_2 \cdot (r_1 \cdot q_3 + r_2 \cdot q_4) \end{aligned} \tag{10}$$

Next let us calculate user strategy for the case of $q_1 = q_3 = q$ and $q_2 = q_4 = 1-q$, i.e., the probability for the attacker choosing an identity is constant if it is discovered. As we have only two identities in this example, the user decision can be modeled as $r_1 = r, r_2 = 1-r$, leading to:

$$\begin{aligned} E_w[u_n] =& P_1 \cdot (1-P_2) \cdot r \cdot q \\ &+(1-P_1) \cdot P_2 \cdot (1-r) \cdot (1-q) \\ &+P_1 \cdot P_2 \cdot (r \cdot q + (1-r) \cdot (1-q)) \end{aligned} \tag{11}$$

This can be further simplified to:

$$E_w[u_n] = \underbrace{(P_1 \cdot q - P_2 + P_2 \cdot q)}_{\text{A}} \cdot r + P_2 - P_2 \cdot q \tag{12}$$

Eq. (12) reveals advised user strategies. As it is a linear function of r, thus the minimum points can be calculated depending on A: it is either at $r = 0$ if $A > 0$, at $r = 1$ if $A < 0$, or at any points if the function is constant ($A = 0$). The latter case means that regardless of defense strategy there is no privacy breach. For example this happens if $q = 0 \wedge P_2 = 0$, i.e., $v_{n\setminus 2}$ cannot be found but the attacker never chooses $v_{n\setminus 1}$. Two similar cases exist: $P_2 = 0 \wedge (q = 0 \vee P_1 = 0)$, and $P_1 = 0 \wedge q = 1$.

Given the calculation above, the user can compute her strategy for setting r if she knows (or at least have an approximation) of the parameters.

5.4.3 Example 2: Minimizing Cost Against Naive Attackers

Now consider a naive attacker and a user having two identities ($y = 2$) in a single dataset. For this case we give the example $\mathbf{q}_m$ vectors in Table 3 describing the naive attacker decisions. Here, for the sake of simplicity, we assumed that $P_1 + P_2 \leq 1$, but otherwise we could have used $\frac{P_1}{P_1+P_2}$ and $\frac{P_2}{P_1+P_2}$.

Modeling user decisions as $r_1 = r, r_2 = 1-r$ the cost of privacy loss can be given as:

$$\begin{aligned} E_w[u_n] =& \underbrace{(P_1 - P_2) \cdot (1 + P_1 \cdot P_2)}_{\text{B}} \cdot r + P_2 - P_1 \cdot P_2 \\ &+ P_1 \cdot P_2^2 \end{aligned} \tag{13}$$

The sign of B depends only on $P_1 - P_2$, as the second term is always positive. Thus when $P_1 > P_2$ the minimum point is at $r = 0$ and the sensitive information should be always assigned to $v_{n\setminus 2}$. For $P_1 < P_2$ it should be assigned to $v_{n\setminus 1}$ ($r = 1$). Strategies proposed by the model follows the common sense again: hide he information in the identity that is harder to be recovered.

Let us take another simple example where the attacker decision is made accordingly to a coin flip in the case of $\mathbf{m} = [1\ 1]$. This modifies Table 3 as $\mathbf{q}_{[1\ 1]} = [0.5\ 0.5]$. Here the expected cost of privacy loss is as follows:

$$E_w[u_n] = \underbrace{(P_1 - P_2)}_{\text{C}} \cdot r + P_2 - \frac{1}{2} \cdot P_1 \cdot P_2 \tag{14}$$

Having term C, the decision cases are the same as in the previous example with B.

5.5 Most Likely Scenario: Attacker Strategy Unknown

In case of the k-anonymity setting, ideally the expected privacy loss is

$$E_k[u_n] = \frac{u_n^-}{k}, \tag{15}$$

as according to the k-anonymity definition there should be at least k entities with the same quasi-identifier (including the user). However, in Section 5.6 we discuss why we have to deal with less favorable cases often.

Now, let us seek an appropriate user strategy for the y-identity model against unknown attackers. From this strategy, we can reasonably expect at least a similar level of expected privacy loss compared to k-anonymity. In order to have that, we propose to use the equilibrium strategy $r_i = \frac{1}{y}$, and show that it is sufficient.

Theorem 3 *Given the attacker model but with no restrictions to the attacker type, using $r_i = \frac{1}{y}$ $(\forall i)$ as a mixed strategy has a threshold for the expected privacy loss as*

$$E[u_n] \leq \frac{u_n^-}{y}.$$

Proof In order to satisfy the theorem, the following criteria needs to be satisfied for strong and weak type of attackers:

$$E_s[u_n] \leq \frac{u_n^-}{y} \text{ and } E_w[u_n] \leq \frac{u_n^-}{y}. \tag{16}$$

The expected privacy loss in case of strong attackers can be easily calculated, and it satisfies this criteria as it is:

$$E_s[u_n] = \frac{u_n^-}{y}. \tag{17}$$

Let us check the expected privacy loss for weak attackers by substituting $r_i = \frac{1}{y}$ to Eq. (7):

$$E_w[u_n] = \left(\sum_{\forall \mathbf{m}} \left(\prod_{\forall j} ((1-m_j) + (-1)^{(1-m_j)} \cdot P_j) \cdot \sum_{\forall i} q_i^{\mathbf{m}} \right) \right) \cdot \frac{u_n^-}{y} \tag{18}$$

However, due to $\sum q_i^{\mathbf{m}} \leq 1$, an upper estimate can be given when $\sum q_i^{\mathbf{m}} = 1$:

$$E_w[u_n] \leq \frac{u_n^-}{y} \cdot \sum_{\forall \mathbf{m}} \left(\prod_{\forall j} ((1-m_j) + (-1)^{(1-m_j)} \cdot P_j) \right) \tag{19}$$

Due to the construction of $\mathbf{m}$, the sum adds up all possible combinations of P_j and $(1-P_j)$ $(\forall j)$, which eventually sums up to 1. Therefore we have:

$$E_w[u_n] \leq \frac{u_n^-}{y}, \tag{20}$$

which, with Eq. (17), satisfies the criteria for the theorem according to Eq. (16). ■

Theorem 3 drives us to interesting conclusions. It shows that despite generally pure strategies are proposed in case of weak attackers, it is yet worth following the equilibrium strategy proposed against strong attackers: then the expected privacy loss would be at most as high as for k-anonymity.

5.6 Comparison with k-anonymity and Risk Mitigation

In the k-anonymity model, there are k structurally identical users, therefore one could be chosen with probability $\frac{1}{k}$. While the expected loss of the y-identity model can be upper bounded with the expected loss for k-anonymity, the risks are not equal for all identities of the user. Let us demonstrate this on a simple example. Let suppose there is a user who has $y = 5$ identities, and $\forall i \leq 4 : P_i \ll P_5$. The user then randomly assigns the sensitive attribute to one of the identities with $r = \frac{1}{5}$. However, while $v_{n\setminus 5}$ has the same chance r as the other identities, getting $v^\star_{n\setminus 5}$ is risky: it is very likely that even a naive attacker would compromise the privacy of the user v_n.

These kind of risks can be easily mitigated if P_i values are known. For instance by creating multiple identities for artificially establishing k-anonymity with a lower k setting for the related identities. This could mean doubling the identity for $v_{n\setminus 5}$ by introducing a structurally equivalent $v_{n\setminus 6}$ but with different sensitive attribute.

Or, if possible, k-anonymity could be established by aligning partial identities to the neighborhood. The difference here to Algorithm 1 is that parameter c is a constraint defined by the neighborhood, not chosen by the user. Fortunately, according to our measurements described in Section 5.7 simply using a high number of identities can help to keep all P_i values significantly low.

However, there is a serious problem with the k-anonymity model that we managed to eliminate in our y-identity model. As in the k-anonymity model it is not the user who controls sensitive values, this can cause problems. For example, if there are m users in the k set with the same sensitive attribute, the probability of privacy loss increases from $\frac{1}{k}$ up to $\frac{m}{k}$. Generally speaking, the if the attacker has an apriori general distribution on the possible values of the sensitive value, this can be fine-tuned by the distribution observed in the k set of users. In the y-identity model the user is allowed to set an arbitrary distribution for the sensitive values where such problems can also be taken into account.

5.7 Evaluating the Predictability of Node Discovery

Here we present a method for estimating the discovery probabilities of nodes. We work with a slightly modified version of Nar09 that eventually provides lower estimates of discovery probabilities. The drawback of Nar09 is that it can only assign a single identity of $v_{n\setminus i} \in G_{tar}$ to $v_n \in G_{src}$ as a match, and according to our measurements, the algorithm is quite deterministic in this: if it gives $\mu(v_n) = v_{n\setminus i}$ once, then it will yield the same match with high probability in subsequent runs. Therefore we would not have any information on the finding probability of other identities.

In order to solve this problem, we committed the following modification. We iteratively run measurements for $\forall v_{n\setminus i} \in \lambda_G(v_n)$. In each round we removed $\forall j \neq i : v_{n\setminus j}$, and then run Nar09 10 times and accumulate node discovery scores as $S(v_{n\setminus i}) = \sum s(v_n, \mu)$. This resulted in an accurate lower estimation how easily each identity can be found; obviously, this can be topped by future algorithms or attackers using a wider range of auxiliary information than topology.

In our experiments, we run these measurements on perturbed datasets of two types (derived for all three networks). In the first case we applied the basic model with $y = 2$ (uniform edge sorting), and in the second case we applied the best model with $y = 5$ (random deletion). We used $V_{ids} = 0.1$. Next we randomly selected 100 nodes from all six datasets having exactly $y = 2$ or $y = 5$, and run the aforementioned simulations regarding the selected nodes. Our results are summarized on Fig. 3.

We used `random.25` and the `top` seeding methods for the re-identification of users having two identities. Fig. 3a shows that results depend on the seed method, and the `top` method produced more consistent results, resulting in more cases when both identities were always found (14.33% of all). While the `random.25` method had less of such cases (12.6%), it was able to find both identities for more nodes, but not consistently (17.6%). All in all, identity separation could be reversed approx. 15% of all cases, which ratio worth considering.

The best model setting with $y = 5$ provided more privacy friendly results. The modified Nar09 (initialized with `random.25`) could correctly re-identify identities only in 7.3% of all cases, and in 2% re-identifications were false matches. We observed no mixed cases having some identities correctly, and others falsely identified. These results shed light on the reason behind why identity separation with 5 identities produced good results in previous measurements: these cases have low re-identification rates and even if there is correct one, only a few of the identities are likely to be found. To be exact, the probability that a partial identity was found at least once was 2.83% ($S(v_{n\setminus i}) > 0$), and only 1.72% of identities was always found ($S(v_{n\setminus i}) = 10$).

Measurements of these kinds are appropriate for estimating the probability of re-identification for identities. However, there are two problems: the user rarely knows the whole network, and results can also depend on the used seeding and attack method, which cannot be certainly known apriori. It would be rational to limit the required user knowledge to a two-hop neighborhood; however, using only such a limited knowledge, we managed to succeeded in approximating these probabilities only in small networks (e.g., few thousand nodes), which we would not consider lifelike. Fortunately, one does not need to know these probabilities in order to have significant protection. Our results indicate that using five identities is strong enough against naive attackers, and using the strategy proposed in Section 5.5 should be adequate in other cases.

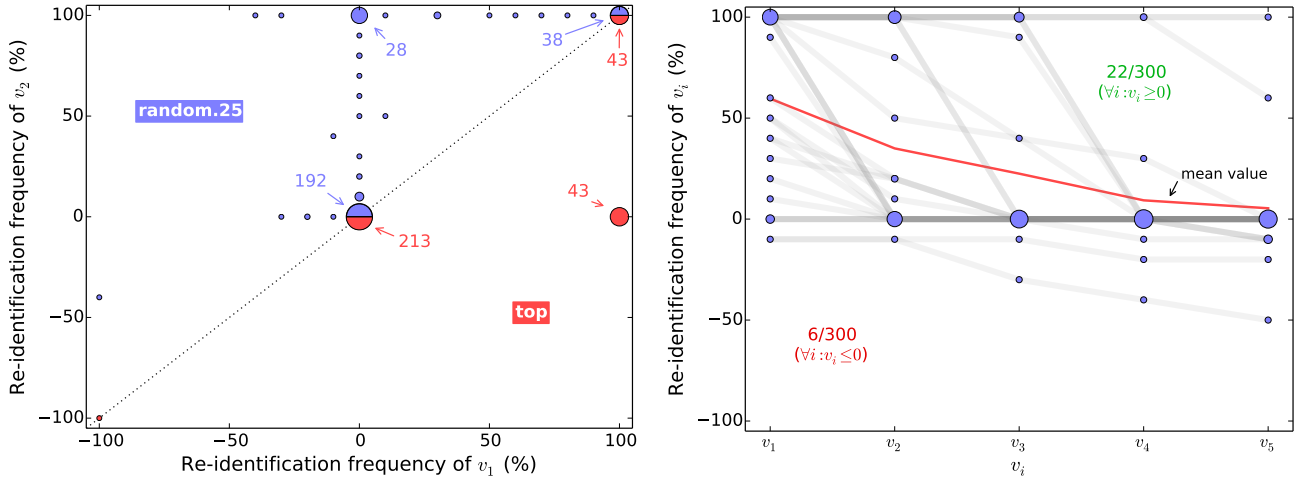
Finally, these measurements are also interesting from an adversarial point of view, too. Theoretically, the attacker can also run a similarly modified version of Nar09 in order to find all partial identities: after finding an identity, it is removed from the network, and the attack is run again. This could be iterated until there is no match for the selected node. After finding all such matches identity separation could be (partially) reversed. According to our measurements (shown on Fig. 3) this can be done only to a very small fragment of the nodes using identity separation with Nar09, but this finding can open an interesting line of future work.

6 Applications of Results

Overall, we believe our results fill a significant gap in the research of protecting user privacy against structural re-identification attacks. There are very few contributions proposing (and analyzing) user centered approaches, that could be applied to existing services. However, adopting the proposed identity separation strategies manually is difficult, and users cannot be expected to manage several partial identities on their own. Therefore, our results should be used to formulate the core principles for designing privacy-enhancing identity management systems that support user behavior in social (or related) services. As there are also several technical issues to be handled in parallel, identity management should be supported by an *identity management tool*, implemented as a browser extension or as a standalone application.

For online social networks one could think of a browser extension as the IDM tool. On a unified user interface, this tool should provide parallel logged-in sessions with different social profiles to support identity separation; thus for changing identity the user would not need to be asked to log out and log in again. Parallelism could be achieved by building on anonymous web browsers [19, 41] in case of web-based social networks. Each separated identity handled by this software would be shared with a group of contacts, where the identity management software sorted connections between different profiles. Our results in this paper provide guidelines how contacts should be handled; e.g., some contacts should be made hidden, while others could be added to multiple profiles.

On the user interface, contacts should appear in their own group (e.g., similar to circles in Google+). Groups can be either related to a single separated identity or consisting multiple of such identities. In the latter case, grouping multiple identities can improve user experience and ease use in general, while having multiple identities can increase the level of privacy. However, as separated identities are quasi fake accounts, users need to exchange valid profile information to maintain



(a) We measured re-identification frequency by initializing with the `random.25` and the `top` methods. The figure shows that results depend on the seed method used by the attacker, as in the case of the `top` method re-identification rates were higher and results were more consistent. As it is shown, identity separation could be reversed certainly only in less than 15% of all cases.

(b) We used `random.25` seeding on the datasets with $y = 5$. Nar09 could re-identify correctly identities only in 7.3% of all cases (with no error), and in 2% re-identifications were false matches (with no correct ones). The figure shows results having the values in the score vector in a descending order; corresponding values are connected with lines. Marker sizes are proportionate to the number of cases we had.

Fig. 3: Results for finding partial identities. In both cases 100 identities were selected from the Epinions, Slashdot and the LJ66k networks having (a) $y = 2$ and (b) $y = 5$ separated identities. The figures indicate the relative frequency of finding each identity.

the functionality of social networks: undercover identities should reveal themselves to contacts to ease communication. The IDM tool could do this automatically by using cryptographic protocols between users who use the same software.

7 Conclusions and Future Work

In this paper, we provided details on how identity separation can be used to tackle re-identification in social networks effectively. Naively using identity separation and assigning a sensitive value to one of the new identities (without further consideration) cannot provide any assurance whether the attacker would find it or not. As a possible additional improvement, we analyzed an applied variant of k-anonymity, and found that this model cannot be implemented effectively in the current context due to the diversity of network structure. As an alternative, we proposed the y-identity model, which introduced several improvements compared to k-anonymity, beside the fact that it can be applied effectively within the context of our discussion. We introduced a reasonable attacker model for the problem, and proved that even if the attacker type is not known (as it happens in real life) and the user acts according to the proposed strategy, the expected privacy loss will be lower or equal compared to the case when k-anonymity can be ideally applied. We additionally discussed that the y-identity model fixes a serious vulnerability of k-anonymity.

We found multiple issues in addition that could be interesting as future work. For example, it might be interesting to consider the re-identification algorithm as a part of the decision making process, and to see how the whole process could be analyzed as a game. We find the development of re-identification algorithms to be the most interesting new line of research: how new algorithms, or variants of existing ones could be used to re-align separated user identities.

Our threat model in this paper was restricted to have regular social networks as the background knowledge of the attacker. However, theoretically an attacker could obtain background knowledge that contains identity separated users, which he could use to reveal hidden attributes in the identity separated anonymous network. Fortunately, if the user adopts identity separation in the right way, the attacker success can be limited even in this case: the adversary could only link a partial identity to the anonymous identity. This could mean no privacy harm if the partial identity was run under a pseudonym. We must note that the y-identity model proposed in this paper provides feasible protec-

tion for such scenarios also. Here the future work should focus on finding appropriate strategies for using non-cooperative identity separation in order to prevent leaks when the attacker background knowledge have identity separation, too.

Acknowledgments

The authors would like to thank Levente Buttyán, István Vajda and Benedek Simon for the fruitful conversations. We are very grateful for Márk Félegyházi and Tamás Holczer for reviewing draft versions of this paper, and for engaging us in meaningful discussions. We would like to also thank the useful comments and suggestions of Gergely Biczók, and also for reviewing the y-identity model in details. The authors are thankful for the comments and suggestions provided by the members of the CrySyS reading group.

References

1. Aggarwal, C.C.: On k-anonymity and the curse of dimensionality. In: Proceedings of the 31st International Conference on Very Large Data Bases, VLDB '05, pp. 901–909. VLDB Endowment (2005). URL http://dl.acm.org/citation.cfm?id=1083592.1083696
2. Assam, R., Hassani, M., Brysch, M., Seidl, T.: (k, d)-core anonymity: Structural anonymization of massive networks. In: Proceedings of the 26th International Conference on Scientific and Statistical Database Management, SSDBM '14, pp. 17:1–17:12. ACM, New York, NY, USA (2014). DOI 10.1145/2618243.2618269. URL http://doi.acm.org/10.1145/2618243.2618269
3. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: Proceedings of the 16th international conference on World Wide Web, WWW '07, pp. 181–190. ACM, New York, NY, USA (2007). DOI 10.1145/1242572.1242598. URL http://doi.acm.org/10.1145/1242572.1242598
4. Bartunov, S., Korshunov, A., Park, S.T., Ryu, W., Lee, H.: Joint link-attribute user identity resolution in online social networks. In: Proceedings of the sixth Workshop on Social Network Mining and Analysis (2012)
5. Beato, F., Conti, M., Preneel, B.: Friend in the middle (fim): Tackling de-anonymization in social networks. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on, pp. 279–284 (2013). DOI 10.1109/PerComW.2013.6529495
6. Beato, F., Kohlweiss, M., Wouters, K.: Scramble! your social network data. In: S. Fischer-Hbner, N. Hopper (eds.) Privacy Enhancing Technologies, *Lecture Notes in Computer Science*, vol. 6794, pp. 211–225. Springer Berlin Heidelberg (2011). DOI 10.1007/978-3-642-22263-4_12. URL http://dx.doi.org/10.1007/978-3-642-22263-4_12
7. Benedek, S., Gulyás, G.G., Imre, S.: Analysis of grasshopper, a novel social network de-anonymization algorithm. Periodica Polytechnica Electrical Engineering and Computer Science **58**(4), 161–173 (2014)
8. Cecaj, A., Mamei, M., Bicocchi, N.: Re-identification of anonymized cdr datasets using social network data. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on, pp. 237–242 (2014). DOI 10.1109/PerComW.2014.6815210
9. Chen, D., Hu, B., Xie, S.: De-anonymizing social networks. Tech. rep., Stanford University (2012)
10. Cheng, J., Fu, A.W.c., Liu, J.: K-isomorphism: Privacy preserving network publication against structural attacks. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, pp. 459–470. ACM, New York, NY, USA (2010). DOI 10.1145/1807167.1807218. URL http://doi.acm.org/10.1145/1807167.1807218
11. Clauß, S., Kesdogan, D., Kölsch, T.: Privacy enhancing identity management: protection against re-identification and profiling. In: Proceedings of the 2005 workshop on Digital identity management, DIM '05, pp. 84–93. ACM, New York, NY, USA (2005). DOI 10.1145/1102486.1102501. URL http://doi.acm.org/10.1145/1102486.1102501
12. Cutillo, L.A., Molva, R., Strufe, T.: Safebook: A privacy-preserving online social network leveraging on real-life trust. Communications Magazine, IEEE **47**(12), 94–101 (2009). DOI 10.1109/MCOM.2009.5350374
13. Goga, O., Lei, H., Parthasarathi, S.H.K., Friedland, G., Sommer, R., Teixeira, R.: Exploiting innocuous activity for correlating users across sites. In: Proceedings of the 22Nd International Conference on World Wide Web, WWW '13, pp. 447–458. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2013). URL http://dl.acm.org/citation.cfm?id=2488388.2488428
14. Gulyás, G.G., Imre, S.: Analysis of identity separation against a passive clique-based de-anonymization attack. Infocommunications Journal **4**(3), 11–20 (2011)
15. Gulyás, G.G., Imre, S.: Measuring local topological anonymity in social networks. In: Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on, pp. 563–570 (2012). DOI 10.1109/ICDMW.2012.87
16. Gulyás, G.G., Imre, S.: Hiding information in social networks from de-anonymization attacks by using identity separation. In: B. Decker, J. Dittmann, C. Kraetzer, C. Vielhauer (eds.) Communications and Multimedia Security, *Lecture Notes in Computer Science*, vol. 8099, pp. 173–184. Springer Berlin Heidelberg (2013). DOI 10.1007/978-3-642-40779-6_15
17. Gulyás, G.G., Imre, S.: Measuring importance of seeding for structural de-anonymization attacks in social networks. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on, pp. 610–615 (2014). DOI 10.1109/PerComW.2014.6815276
18. Gulyás, G.G., Imre, S.: Using identity separation against de-anonymization of social networks. Journal of Transactions on Data Privacy **8**(2), 113–140 (2015)
19. Gulyás, G.G., Schulcz, R., Imre, S.: Comprehensive analysis of web privacy and anonymous web browsers: are next generation services based on collaborative filtering? In: L. Capra, I. Wakeman, N. Foukia, S. Marsh (eds.) Proceedings of the Joint SPACE and TIME Workshops 2008. CEUR Workshop Proceedings (2008)
20. Gulyás, G.G., Schulcz, R., Imre, S.: Modeling role-based privacy in social networking services. In: Emerging Security Information, Systems and Technologies, 2009. SE-

CURWARE '09. Third International Conference on, pp. 173–178 (2009). DOI 10.1109/SECURWARE.2009.34
21. Gulyás, G.G., Schulcz, R., Imre, S.: Separating private and business identities. Digital Identity and Access Management: Technologies and Frameworks: Technologies and Frameworks pp. 114–132 (2012). DOI 10.4018/978-1-61350-498-7.ch007
22. Jain, P., Kumaraguru, P., Joshi, A.: @i seek 'fb.me': identifying users across multiple online social networks. In: Proceedings of the 22nd international conference on World Wide Web companion, WWW '13 Companion, pp. 1259–1268. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2013). URL http://dl.acm.org/citation.cfm?id=2487788.2488160
23. Ji, S., Li, W., He, J., Srivatsa, M., Beyah, R.: Poster: Optimization based data de-anonymization (2014). Poster presented at the 35th IEEE Symposium on Security and Privacy, May 18–21, San Jose, USA
24. Ji, S., Li, W., Mittal, P., Hu, X., Beyah, R.: Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization. In: Proceedings of the 24th USENIX Conference on Security Symposium, SEC'15, pp. 303–318. USENIX Association, Berkeley, CA, USA (2015). URL http://dl.acm.org/citation.cfm?id=2831143.2831163
25. Ji, S., Li, W., Srivatsa, M., Beyah, R.: Structural data de-anonymization: Quantification, practice, and implications. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, CCS '14, pp. 1040–1053. ACM, New York, NY, USA (2014). DOI 10.1145/2660267.2660278. URL http://doi.acm.org/10.1145/2660267.2660278
26. Ji, S., Li, W., Srivatsa, M., He, J.S., Beyah, R.: Information Security: 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings, chap. Structure Based Data De-Anonymization of Social Networks and Mobility Traces, pp. 237–254. Springer International Publishing, Cham (2014). DOI 10.1007/978-3-319-13257-0_14. URL http://dx.doi.org/10.1007/978-3-319-13257-0_14
27. Korula, N., Lattanzi, S.: An efficient reconciliation algorithm for social networks. Proc. VLDB Endow. **7**(5), 377–388 (2014). DOI 10.14778/2732269.2732274. URL http://dx.doi.org/10.14778/2732269.2732274
28. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: In ICDE (2007)
29. Narayanan, A., Shi, E., Rubinstein, B.I.P.: Link prediction by de-anonymization: How we won the kaggle social network challenge. In: The 2011 International Joint Conference on Neural Networks, pp. 1825–1834 (2011)
30. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: Security and Privacy, 2009 30th IEEE Symposium on, pp. 173–187 (2009). DOI 10.1109/SP.2009.22
31. Nilizadeh, S., Kapadia, A., Ahn, Y.Y.: Community-enhanced de-anonymization of online social networks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, CCS '14, pp. 537–548. ACM, New York, NY, USA (2014). DOI 10.1145/2660267.2660324. URL http://doi.acm.org/10.1145/2660267.2660324
32. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT press (1994)
33. Parker, J.: What nsa's prism means for social media users. http://www.techrepublic.com/blog/tech-decision-maker/what-nsas-prism-means-for-social-media-users/. Accessed: 2014-05-26
34. Pedarsani, P., Figueiredo, D.R., Grossglauser, M.: A bayesian method for matching two similar graphs without seeds. In: Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on, pp. 1598–1607 (2013). DOI 10.1109/Allerton.2013.6736720
35. Peng, W., Li, F., Zou, X., Wu, J.: Seed and grow: An attack against anonymized social networks. In: Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on, pp. 587–595 (2012). DOI 10.1109/SECON.2012.6275831
36. Pham, H., Shahabi, C., Liu, Y.: Ebm: an entropy-based model to infer social strength from spatiotemporal data. In: Proceedings of the 2013 international conference on Management of data, pp. 265–276. ACM (2013)
37. Stanford network analysis platform (snap). http://snap.stanford.edu/. Accessed: 2014-04-22
38. Srivatsa, M., Hicks, M.: Deanonymizing mobility traces: using social network as a side-channel. In: Proceedings of the 2012 ACM conference on Computer and communications security, CCS '12, pp. 628–637. ACM, New York, NY, USA (2012). DOI 10.1145/2382196.2382262. URL http://doi.acm.org/10.1145/2382196.2382262
39. Sweeney, L.: Uniqueness of simple demographics in the us population. Tech. rep., Technical report, Carnegie Mellon University (2000)
40. Sweeney, L.: K-anonymity: A model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(5), 557–570 (2002). DOI 10.1142/S0218488502001648. URL http://dx.doi.org/10.1142/S0218488502001648
41. Tor browser. https://www.torproject.org/projects/torbrowser.html.en. Accessed: 2014-11-04
42. Yartseva, L., Grossglauser, M.: On the performance of percolation graph matching. In: Proceedings of the First ACM Conference on Online Social Networks, COSN '13, pp. 119–130. ACM, New York, NY, USA (2013). DOI 10.1145/2512938.2512952. URL http://doi.acm.org/10.1145/2512938.2512952
43. Zou, L., Chen, L., Özsu, M.T.: K-automorphism: A general framework for privacy preserving network publication. Proc. VLDB Endow. **2**(1), 946–957 (2009). DOI 10.14778/1687627.1687734. URL http://dx.doi.org/10.14778/1687627.1687734

Algorithm 1: $(k, 2)$-anonymity with edge modification

```
1: procedure K-AnonymizeNode(G, v_i, c, k)
2:     V_i ← G.nbrs(v_i)
3:     V_i^2 ← G.nbrs(V_i) \ {v_i}
4:     c' ← c, V_k ← {}, E_k ← {}
5:     while c' ≥ 1 and |V_k| = 0 do
6:         κ ← {}          ▷ Groups having c' common neighbors with v_i
7:         for all v_j ∈ V_i^2 do
8:             V_{i∩j} ← V_i ∩ G.nbrs(v_j)
9:             if |V_j| = c and |V_{i∩j}| = c' then
10:                κ[V_{i∩j}] ← κ[V_{i∩j}] ∪ {v_j}
11:            end if
12:        end for
13:        for all κ[V_{i∩j}] if |κ[V_{i∩j}]| ≥ k − 1 do
14:            if c = c' then      ▷ k-anonymity without modification
15:                V_k ← κ[V_{i∩j}]
16:                break
17:            end if
18:            ψ ← {}          ▷ Get new neighbors related to the k-group
19:            for all v_j ∈ κ[V_{i∩j}] do
20:                V_{j\i} ← G.nbrs(v_j) \ V_i \ κ[V_{i∩j}] \ {v_i}
21:                for all v_l ∈ V_{j\i} do
22:                    ψ[v_l] ← G.nbrs(v_l) ∩ κ[V_{i∩j}]
23:                end for
24:            end for
25:            η ← {}          ▷ Filter applicable groups and neighbors
26:            for all ψ[v_l] do
27:                for all γ ⊆ ψ[v_l] if |γ| = k − 1 do
28:                    η[γ] ← η[γ] ∪ {v_l}
29:                end for
30:            end for
31:            if ∃η[γ] that |η[γ]| ≥ c − c' then
32:                pick η[γ] where |η[γ]| ≥ c − c'
33:                V_k ← γ
34:                E_k ← η[γ]
35:                break
36:            end if
37:        end for
38:        c' = c' − 1
39:    end while
40:    return V_k, E_k   ▷ Existing and new neighbors for k-anonymity
41: end procedure
```