MŰEGYETEM 1782

Budapest University of Technology and Economics

Faculty of Electrical Engineering and Informatics

Department of Networked Systems and Services

Mobile Communications and Quantum Technologies Laboratory (MCL),

and

Laboratory of Cryptography and System Security (CrySyS)

Protecting Privacy Against Structural De-anonymization Attacks in Social Networks

Ph.D. Dissertation

of

Gábor György Gulyás

Research Supervisor:

Sándor Imre, DSc.

2015

Alulírott *Gulyás Gábor György* kijelentem, hogy ezt a doktori értekezést magam készítettem, és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

I, the undersigned *Gábor György Gulyás* hereby declare that this Ph.D. dissertation was made by myself, and I only used the sources given at the end. Every part that was quoted word-for-word, or was taken over with the same content, I noted explicitly by giving the reference of the source.

Budapest, 2015. február 9.

. . . .

Gulyás Gábor György

Abstract

Privacy is a core concern of online social networks. Probably the corner stone challenge is the amount of identifiable information contained in social network meta-data: the underlying graph structure. Sanitized social network information is occasionally shared with third parties, such as business partners and researchers. Previous research developed de-anonymization attacks that can re-identify social network users in such datasets by using public data sources, e.g., obtained by crawling other networks. A strong class of such attacks considered in this dissertation solely consider structural information of the social graph, and achieve large-scale re-identification.

This yields the need for solutions protecting user privacy in social networks. In this thesis, I consider client-side solutions that involve users only, and can be adopted gradually within existing services. Specifically, I investigate the use of an identity management technique called identity separation as a tool for tackling de-anonymization attacks, and analyze several settings of the technique. Initially, my experiments focus on measuring the effectiveness of basic, non-cooperative identity separation mechanisms. Then, I experimentally check if multiple cooperation models can improve overall protection. Finally, I evaluate several strategies where the focus is on protecting the individual privacy of participants. Some of these strategies provide feasible protection in case of the state-of-the-art attack, while others have theoretical guarantees.

Besides, I also contribute to the analysis of attack algorithms: I propose methods for measuring anonymity, and characterize how the initialization of these algorithms can affect the overall performance of the attack.

Kivonat

A pivátszféra védelme kiemelt fontosságú kérdés a közösségi hálózatokban. Az egyik legnagyobb kihívást talán a közösségi hálózatok alapját adó gráf struktúra adja, amely lehetőséget kínál a felhasználók kéretlen azonosítására. Számos esetben előfordul, hogy a közösségi hálózatok üzemeltetői megosztanak anonimizált adatokat harmadik felekkel is, mint például kutatókkal vagy üzleti partnerekkel, azonban léteznek olyan ún. de-anonimizációs algoritmusok, amelyek segítségével újraazonosíthatóak az ezekben lévő felhasználók, például nyilvánosan elérhető szolgáltatásokból begyűjtött információk segítségével. A disszertációmban ezen támadások egy erőteljesebb típusát vizsgálom, amely csak a közösségi hálózat struktúrális információt használja, és képes a felhasználók nagy léptékű újraazonosítására.

Ezért szükség van privátszérát erősítő megoldásokra. Disszertációmban olyan kliensoldali megoldást vizsgálok, melynek alkalmazása csupán a felhasználón múlik, és nem muszáj a védekezni kívánó felhasználóknak új szolgáltatásba migrálniuk: lehetőség van fokozotos átállásra a meglévő szolgáltatásokban is. Egész pontosan az identitásmenedzsment módszerek közé tartozó ún. identitás szeparáció elnevezésű – mint egy, a de-anonimizáció elleni védekezési módszert – javaslok megoldásként, és ennek számos alkalmazási módját vizsgálom munkámban. Kezdeti kísérleteimben az együttműködés nélküli, egyszerűbb identitás szeparációs stratégiák hatékonyságét mérem, majd megvizsgálom, hogy különféle kooperációs stratégiák alkalmazásával javíthatóak-e az eredmények. Végezetül az egyéni védelemre összpontosító stratégiákat vizsgálom, amelyek között van olyan, amelyik a jelenleg legkorszerűbb támadással szemben ad megfelelő védelmet, illetve van olyan is, amelyik elvi korlátot ad még erősebb támadóval szemben is.

Mindemellett dolgozatom a támadási módszerek elemzésével kapcsolatban is szolgál új eredményekkel: javaslok módszereket a felhasználói anonimitás mérésére, illetve megmutatom, hogy a támadások inicializálása hogyan hat ki a támadás egészének sikerességére.

Acknowledgements

> 'It is the glory of God to conceal a thing; But the glory of kings is to search out a matter.' (Proverbs 25:2, American Standard Version)

Primarily, I would like to thank Sándor Imre for his supervision of my research, his support and encouragement in all times. He always welcomed me with open doors for discussing research issues, and his positive attitude helped me a lot on my journey as a PhD student.

I would like to also thank Levente Buttyán for giving me the opportunity to work in an inspiring work environment in the prestigious CrySyS Lab, and for pushing me forward in the last two years of thesis work.

I am also thankful for the continuous support of Iván Székely, who initially motivated me to research Privacy-Enhancing Technologies back in 2005.

I would like to highlight and thank the inspiring discussions we had with Tamás Holczer, and Márk Félegyházi; these were very useful. I am grateful for Tamás Holczer and Máté Horváth for reviewing draft parts of my dissertation. I also had fruitful conversations with András Telcs, who provided useful comments for multiple of my works.

I am also thankful for the join efforts and publications with several colleagues: Benedek Simon, Károly Boda, Ádam Máte Földes, Róbert Schulcz, Tamás Besenyei, Tamás Paulik, Sándor Dargó, András Kóbor. I am also grateful to the current and former members of the CrySyS and MCL Labs.

I would like to thank the work of reviewers of the first version of my thesis, Julien Freudiger and Gergely Biczók. Their constructive suggestions, the large number of useful comments, and our discussions greatly contributed to improve my work.

Hereby I would like to acknowledge gratefully the PhD student scholarship provided by the High-Speed Networks Laboratory. Partial support of my research are also appreciated for the Mobile Innovation Centre, the BME-Infokom Innovátor Nonprofit Kft.

I would like to thank my wife Orsi for her love, support and patience. Her help provided me a stable background. Finally, I want to thank my family and friends for supporting me.

Contents

Acronyms

DBLP Abbreviation I use in general for the derivatives of the DBLP network.

EP Abbreviation I use in general for the derivatives of the Epinions network.

FB Abbreviation I use in general for the derivatives of the Facebook network.

FiM Friend-in-the-middle model [1].

LTA Local Topological Anonymity

LJ Abbreviation I use in general for the derivatives of the LiveJournal network.

LCC Local Clustering Coefficient

MAC Media Access Control (address)

Nar09 Structural re-identification algorithm proposed by Narayanan and Shmatikov in 2009 [2].

PET Privacy-Enhancing Technology

PIDM Privacy-Enhancing Identity Management

PKC Abbreviation I use in general for the derivatives of the Pokec network.

PRIME Privacy and Identity Management for Europe

PrimeLife Successor project of PRIME.

RBP Role-Based Privacy

SNAP Stanford Network Analysis Project

SD Abbreviation I use in general for the derivatives of the Slashdot network.

WV Abbreviation I use in general for the derivatives of the Wikivote network.

Chapter 1

Introduction and Overview

Social media services are used every day by hundreds of millions, or even more. However, beside the values these services give to humanity, social media also serves as an optimal platform for all kinds of surveillance activities, as members can snoop upon each other, commercial parties can buy vast amounts of private data, and as recent events confirm [3], government surveillance is also present as well. Social networks are definitely one of the key ingredients in shaping our societies today, accelerating the shift from information societies to surveillance societies [4].

Due to the myriad of related privacy problems [5, 6], a large number of privacy-enhancing technologies (PETs) have been proposed. One of the most challenging tasks is to make identification by relationship information cumbersome, or even impossible. There are solutions aiming to solve this by proposing replacement of centralized social networks with distributed platforms. Other works propose to modify the functionality social networks in fundamental ways, eventually requiring the migration of users to novel services to maintain their privacy (e.g., such as Diaspora [7]). Another line of research constructs techniques that could be put into use by social network providers to release meaningful but still private data, e.g., by using differential privacy [8].

However, we need solutions that can be adopted gradually, thus allow contacting others who have not yet taken steps to strengthen their privacy, but yet enhance the users' privacy. As large social network providers can be forced to handle user data to governments or sell user data to third parties, the control of anonimization need to lie in the hand of the users.

In addition, there are several systems, where connections between entities are not considered as an explicit feature, while this kind of meta-data yet provides means of identification. Such attacks have been demonstrated for location privacy, where it has been shown that co-location information in spatio-temporal dataset can be used to reconstruct the underlying social network, and finally structural information crawled from social networks can be used to identify users [9–12]. These and similar cases yield for solutions described above, where the privacy control lies in the hands of users.

1.1 Motivation

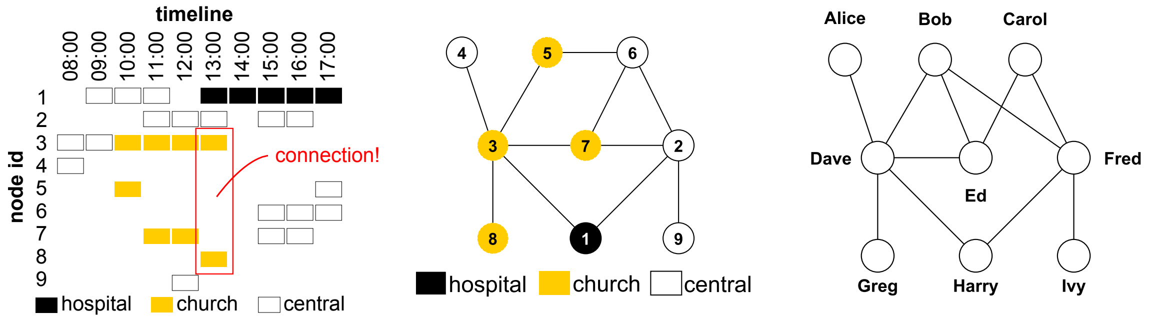
Datasets are usually protected by naive anonimization when shared with business or research partners: explicit identifiers are removed (such as names, user ids or email addresses), and the graph structure is slightly perturbed (e.g., a small fraction of edges are removed or added). Unfortunately, naive data anonymization techniques cannot provide an acceptable level of protection, as several works have proven that nodes in sanitizated datasets can be re-identified with high accuracy [2, 10, 12–20]. Most of these methods are capable of achieving large-scale re-identification of social datasets consisting even of hundred thousand records or more.

In particular, I consider a strong class of attacks, where de-anonymization is executed by using structural information only [2,10,12–16]. The following example demonstrates the core principles of these attacks, when identities that were not present in the original datset are recovered [2, 12]. It also gives an insight of the privacy threat when co-location information in spatio-temporal datasets (like mobility traces or check-ins) are converted into a social network graphs [21] to be re-identified as a social network.

Let us consider an attacker who obtains spatio-temporal data as given in Fig. 1.1a. For example, the attacker could buy this data from a Wifi service provider of a small city, who intentionally collects device identifiers that pass by their access points placed at different locations (e.g., smartphones with Wifi turned on). After buying the dataset, the attacker can create an anonymous social graph as Fig. 1.1b based on the co-occurences of each identifier at the same place and time slot. From a business point of view, the resulting dataset would be even more valuable for the adversary if it could label each node with a publicly known identity.

After crawling social relations from another source, for instance from a publicly available online social networking site (including only users who claim to live in that small city), the re-identification process can be done by the attacker in two steps. The background knowledge, or auxiliary dataset is shown in Fig. 1.1c. First, the attacker can search for nodes with outstanding properties, like using node degree as in this case. By searching for unique, high degree nodes the attacker can create a re-identification match between the nodes $v_{Dave} \leftrightarrow v_3$ and $v_{Fred} \leftrightarrow v_2$. As no more of such mappings can be found, next nodes related to existing mapped ones can be re-identified. For example, v_{Harry} has two connections (which is not unique globally), and he is connected to both v_{Dave}, v_{Fred}; this boils down choices to the re-identification mapping of $v_{Harry} \leftrightarrow v_1$.

After deriving conclusions from the results of the attack, the attacker can now maliciously use the fact that Harry visited the hospital for several hours, such as



(a) Anonymized spatio-temporal data

(b) Underlying social network reconstructed from (a)

(c) Crawled public network (as auxiliary data)

Figure 1.1: For example, an attacker can buy anonymized spatio-temporal data for business analysis (a), from which co-occurences can be used to reconstruct a possible underlying social network (b). Next, structural information crawled from a public social networking site (c) can be used to re-identify nodes in the sanitized dataset.

blackmailing Harry with publishing this information among his friends or employer, or can be used for sending unsolicited advertisements with personally-tailored content.

However, there is a considerable number of similar use case scenarios. For example, such attacks can be used to correlate accounts in two different social networking services, even if users try hiding under different pseudonyms. Or an adversary could buy a sanitized dataset from a social network provider which contains sensitive information (e.g., political or religious preferences), and he could use another social network to reveal the identities within. We could also think of a variety of governmental use cases that involve matching identities between datasets, e.g., call and social networks. In addition, there are several works in the emerging field of re-identification that presented the possibilities of exploiting meta-data for identification, e.g.: BitTorent connections [22], group memberships on social networking sites [23], or Bitcoin transaction history [24].

In order to remedy the present situation, the analysis of a user centered technique is in the focus of my dissertation, called identity separation. This technique could be applied to existing services without modification of the service itself, even without getting the consent of the service provider, and can be deployed gradually. 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 [25]. This can also be applied to social networks to segregate information with different groups of contacts.

Returning to the previous example, identity separation could be applied by using different identifiers in different contexts, e.g., changing the MAC, or using different user names for check-in services. For example, Harry could change his MAC address when arriving at the hospital (or turn off wireless totally), in order to avoid this information being linked to his identity.

1.2 Results

As I mentioned above, my dissertation focuses on the analysis of identity separation, but my work also concerns some important aspects of structural re-identification attacks that have not been addressed previously. My contributions are as follows; I provide references of corresponding publications and sections providing the details.

Analysis of Structural Re-identification Algorithms

- I proposed a family of measures called Local Topological Anonymity (LTA), that enable the relative assessment of the risk of re-identification for a single node. I showed that there is a particular variant called LTA$_A$ which provided values that had strong rank correlation with node re-identification rates for the state-of-the-art and Grasshopper attacks.
  Related publications: [C3, J2, J3], Section 4.1

- I showed that node degree (LTA$_{deg}$) is an efficient, easy to calculate alternative for LTA$_A$. I additionally showed how degree distribution of networks determines which metric should be used for the state-of-the-art attack: LTA$_{deg}$ in networks where the proportion of low degree nodes are relatively high, and LTA$_A$ in others.
  Related publications: [J2, J3], Section 4.1

- For the state-of-the-art algorithm, I characterized the importance of initialization. I showed how the maximum number of re-identified nodes can depend on the seeding method and its parameters. I have characterized how the minimum number of seed nodes depends on network properties and the seeding method. I also characterized seed stability and showed that even an extremely low number of seed nodes can also lead to large-scale propagation.
  Related publications: [C1], Section 4.2

Evaluation of Identity Separation

The concept of how identity separation could be used in social network based services is introduced in [C7, C8], and in my work I used a statistical model capturing possible user behaviors in four sub-models that was originally published in [J4]. These modeling issues are described in Section 3.5.

- I provided the general formula of failure probability of global identification (seeding) when identity separation is used. Using this formula, I elaborated the lower

estimate of failure probability for clique-based seeding, and for a seeding method identifying top degree nodes. I showed with numerical analysis that there are efficient strategies for users to protect themselves with identity separation against these seeding methods.
Related publications: [J4], Section 5.1

- I measured the sensitivity of the propagation phase of the state-of-the-art attack against features of identity separation, and showed the attack is quite robust: a high number of non-cooperating users need to participate to decrease the number of correctly re-identified nodes significantly.
Related publications: [C2, J3], Section 5.2, 5.3.4

- I characterized several properties of non-cooperative identity separation. In particular, I showed that even if the attacker changes the seeding method or seed size, he cannot significantly affect his results against identity separation used in the network.
Related publications: [C2, J2, J3], Section 5.3

- I showed that even for a simple local cooperation scheme, a lower number of participants are enough to defeat re-identification compared to the non-cooperative setting.
Related publications: [J1], Section 5.4

- I showed that by using LTA_A and LTA_{deg} as a global node-selection heuristic for cooperative identity separation, the required number of participants is a small fraction compared to the non-cooperative case. In addition, I showed that changing seeding method or increasing seed set size cannot significantly enhance the attacker's results.
Related publications: [J1–J3], Section 5.5

- I showed that both for non-cooperative and globally cooperative identity separation the participation of top degree nodes is crucial. Without their support, the performance of protection of network privacy degrades rapidly.
Related publications: [C2, J1, J3], Section 5.3.1, 5.6

Evaluation of Individual Strategies

- I showed that even if a handful of users adopt identity separation, their re-identification results stay proportional to measurements observed in networks where strategies are adopted homogeneously. I proposed and successfully evaluated a method of targeted information hiding, that uses decoy identities to compel

the state-of-the-art attack algorithm finding non-relevant information.
Related publications: [C2, J3], Section 6.1, 6.2

- I provided a method for calculating the lower bound of the probability of the discovery of partial identities with a simple modification of the state-of-the-art attack. I showed that even with this modification only a fragment of partial identities can be found and merged.
  Related publications: [J1], Section 6.3

- I proposed $(k,2)$-anonymity, a variant of k-anonymity to be adopted individually for tackling re-identification attacks. By evaluating `K-AnonymizeNode`, an algorithm that sets a $(k,2)$-anonymous setting for a given node, I showed that the concept of k-anonymity cannot be applied efficiently within the current context.
  Related publications: [J1], Section 6.4.1

- I designed the y-identity model as an alternative solution to k-anonymity. I proved that differing strategies are the best against weak and strong attackers. I also proved that the game theoretic equilibrium strategy proposed for strong attackers should be used if the attacker is unknown (i.e., can be either weak or strong), as it has a feasible higher bound on the expected privacy loss.
  Related publications: [J1], Section 6.4.2

1.3 Outline

The outline of the dissertation is as follows. In Chapter 2 I discuss the most relevant related works regarding the topics of structural re-identification, graph privacy protection and privacy-enhancing identity management (covering identity separation). Then, in Chapter 3 I introduce the notation and definition I used in my work, and also how simulation experiments were executed, detailing all important parameters to maintain repeatability.

The main results of my work are presented in Chapter 4, 5, and 6. In Chapter 4 I discuss my general findings that are related to de-anonymization attacks, as evaluation of importance measures and seeding. Then in Chapter 5 I provide the general analysis of identity separation as a possible tool for tackling re-identification attacks. This evaluation contains the detailed analysis of both non-cooperative, locally and globally cooperative identity separation. Then in Chapter 6 I evaluate approaches that focus on maintaining personal privacy, rather than network privacy.

Finally, I discuss how my results could be applied in real life scenarios in Section 7.1, and I specify possible future work task that I find most relevant or scientifically interesting in Section 7.2. Finally, I conclude my work in Chapter 8.

Chapter 2

Related work

> 'They who can give up essential liberty to obtain a little temporary safety deserve neither liberty nor safety.' (Benjamin Franklin)

In this chapter, I discuss the most important works related to the topic of my dissertation which concerns several topics. In Section 2.1 I introduce the main privacy concerns in social networks, and in Sections 2.1.1 and 2.1.2 I discuss the works on structural anonymity and attacks against privacy. In Section 2.2 I discuss relevant privacy-enhancing technologies, and in Section 2.3 I introduce the concept proposed in my dissertation.

2.1 Identification and De-anonymization Algorithms

Social network services bear a great variety of privacy issues since their beginning [6,26]. For example, lack or inconvenience of privacy controls can lead to unwanted information leakage, finally resulting in issues with employment, unintentional fame or stalking by other users. Furthermore, posted content, profile information or relationship information is usually sold to third parties, which also raises privacy concerns. While most of these problems can be handled by self-censoring or using pseudonyms, in my work, I focus on how relationship information, a seemingly anonymous, but sensitive meta-data could be exploited and protected.

There are several scenarios in which such meta-data can be obtained by a malicious party. Social networks occasionally publish or sell naively sanitized social data, where explicit identifiers are removed, and the network structure (or attributes) is slightly perturbed for privacy protection. Occasionally, other types of meta-data can be used to reconstruct social relations between users. For example, there are examples where spatio-temporal data was used to recreate the underlying social network of users, which was then successfully de-anonymized [10,12].

Having an anonymous network, different approaches exist for re-identification. User profiles can be revealed by targeted attacks based on specific profile attributes [19,27]. These attacks are limited in the number of users they can recover, but can work with high accuracy: in their work, Vesdapunt and Garcia-Molina provided an example targeted queries reached accuracy of 88.3% [27]. There are other examples when activities and attributes are used together to correlate user profiles between social networks at a larger scale [20, 28]. However, sometimes no attributes are available for making quasi identifiers, but only structure. Therefore, there is a variety of attacks that consider structural information only for revealing user identities [2,10,12–16], while several other works involve both content, attributes and activites beside structure in the process of re-identification [17,18,29,30]. In my work, I focus on the strong class of attacks using only structural information.

Structural anonymity can be characterized both for the whole network, and for a single node, which can be furthermore specified as global or local identification (i.e., the node is unique globally in the whole network or in the neighborhood of some other nodes). Accordingly, early works focus on how nodes can be identified in their global context, but the paper of Narayanan and Shmatikov in [2] opened up a new line of attacks, where large-re-identification could be achieved efficiently, even for networks sized larger than tens of thousands of nodes. Related results fertilized other lines of research that go far beyond social networks; for example, it has been shown that the algorithm in [2] can be used to re-identify large sets of spatio-temporal data [12], when node relations are mined from co-occurences of nodes.

2.1.1 Initial Works on Structural Anonymity

The work of Sing and Zhan define a network level structural measure, called topological anonymity [31]. Their measure describes variance of patterns in the complete graph with a single, normalized real number, based on node degree anonymity sets. The following other works rather focused on describing global structural anonymity for individual nodes.

The term of structural equivalence appeared in sociology before the research of networks and re-identification attacks [32]. Structural equivalence requires a very high level of similarity, which is also probably rare: such nodes need to have their incoming and outgoing edges to the same nodes.

Practical structural node anonymity measures reflect a node's hiding ability against certain re-identification schemes, mostly based on global structural uniqueness respecting a given fingerprinting scheme or similarity measures. Therefore, a node can be considered anonymous if there are a number of equivalent or sufficiently similar coexisting structural facsimiles present [33–35]. The number of such structural alteregos can represent the level of anonymity in these cases (as they form an anonymity set together).

Several variants of global re-identification techniques are derived from the concept of k-anonymity [36]. Zhou and Pei define that a node is k-anonymous if there are at least (k-1) other nodes with a similar neighborhood (limited to adjacent nodes) [35]. Liu and Terzi define k-degree anonymity similarly: a node is k-degree anonymous, if there are at least (k-1) other nodes with the same degree values in the network [33]. Hey et al. define k-candidate anonymity based on multiple types of node fingerprints: vertex refinement queries (a node is identified by the degree values of its neighbors, or neighbors of neighbors), subgraph queries (a node is identified by a surrounding subgraph described by the implied edges), hub fingerprint queries (a node is identified by its relation to specified hub nodes) [34].

However, structural node anonymity is not explicitly measured in all works. Backstrom et al. present global re-identification phases (an active and a semi-active) where an attacker attempt to inject a unique subgraph in the graph prior to anonymization [13]. Narayanan and Shmatikov in [2] use a uniqueness criterion for 4-cliques for global re-identification of nodes, where the degree and the number common neighbors of clique members are considered for distinguishing the clique structure from others. It is not possible to objectively measure anonymity in these cases as it depends on the background knowledge of the attacker.

2.1.2 Large-Scale Structural Re-identification Attacks

The algorithm proposed of Narayanan and Shmatikov in 2009 (to which I later refer to as Nar09) had a significant novelty compared to the literature discussed so far: it applied local comparison of nodes based on previously discovered matching of neighboring nodes [2]. The Nar09 algorithm aims to reveal the identities of nodes within a sanitized graph (the target graph) by using a social network obtained from an auxiliary source (the source graph). The authors in their main experiment re-identified 30.8% of nodes being mutually present in a Twitter and a Flickr crawl with a relatively low error rate of 12.1%.

Works following their approach also used a similar procedure; in most cases these consist of an initialization phase (or seed phase), which is then followed by a propagation phase. In general, the seeding identifies a small set of globally outstanding nodes, and then the propagation phase extends this set, for instance, by searching locally outstanding nodes that are connected to the set of already re-identified ones. These phases can also be named as global and local re-identification phases.

In their original experiment, the seeding is based on 4-cliques. The steps of the propagation phase are iterated on the neighbors of the nodes already re-identified until new matchings can be discovered (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 exchanged, 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. Further details of the algorithm can be found in the psuedo code provided in Appendix A.1.

Narayanan et al. in 2011 presented another variant of their attack [14] specialized for the task of working on two snapshots of the same network, that could achieve a higher recall rate. Another proposal of Wei et al. [15] challenged Nar09; however, their attack is only evaluated against a light edge perturbation procedure, instead of the more realistic one proposed in [2]. The latter deletes both nodes and edges from both networks (resulting overlaps can be as low as 25%), while in [15] perturbation only adds edges to the target network (up to 3%) without any deletion. For a more comprehensive evaluation, their algorithms need to be compared with a perturbation method that includes deletion. In addition, experiments in [15] are performed on two small graphs consisting only of handful of nodes (graph vertex sizes are 125 and 600) – if it is feasible for the seed-and-grow algorithm, a comparison on larger datasets need to be done.

Pedarsani et al. proposed a novel type of attack that can work without any initial input such as seeds [16]. Their design incorporated seeding into the propagation phase, as the initial propagation step starts identifying top nodes according to a given node fingerprint measure. However, their algorithm requires very high similarity between the source and target datasets (e.g., $\alpha_v = 1.0$ and $\alpha_e = 0.85$; for explanation, see Section 3.3), which is hard to meet in many cases. Additionally, their work was experimentally tested only on a single, small network with 2,024 nodes and 25,603 edges.

Danezis and Sharad presented a generic, machine learning-based deanonymization framework for the evaluation of anonymization schemes [37], which can be trained on a relatively small set of sanitized data. While their results cannot be directly compared to global matching algorithms such as [2], their framework can be used for testing new schemes, such as the one proposed in my dissertation. Using their work to break identity separation goes beyond the scope of this dissertation, and assigned as future work.

It has been shown that even a relatively small amount of mobility data can easily identify users [9], and even short periods of surveillance enable identification [11]. However, it was first shown by Srivatsa and Hicks that location traces can also be re-identified with similar methods what was used for social networks [10]. 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 the ability of algorithms using spatiotemporal data for making social network connections, can be extended to large datasets [21]. Building upon their work, Ji et al. showed that spatiotemporal data at the scale of hundred thousand entities can be easily re-identified [12]: 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.

In [J2] we have proposed Grasshopper, a structural de-anonymization algorithm (later referred as Grh) which has some advanced properties compared to Nar09. It can be initiated with a significantly smaller seed set, and Grh has a negligibly small error rate compared to Nar09. In terms of its operating principles, one of the key improvements of Grh is to use a more complex node comparison method that involves the weighting of existing mappings. While Grh can achieve yield levels higher than Nar09 when the attacker background knowledge is rather noisy (i.e., there is a lower overlap ratio between the auxiliary and target datasets), in other cases correct identification rates can be significantly smaller. For example, in the Epinions dataset when the adversary has a perfect background knowledge, Grh could only correctly re-identify the half of what Nar09 achieved [J2]. Further details of the Grh algorithm is provided with the pseudo code in Appendix A.2.

None of the works appeared since [2] provided algorithms that were proved to be generally advanced alternatives to Nar09 (e.g., having higher re-identification rates generally): some algorithms were proposed to work on different data types (e.g., on location data [10]), others were crafted to work under specific circumstances (e.g., when the attacker knowledge and the target network is quite similar [14,16]), some could only provide better results conditionally (e.g., in case of noisy background knowledge [J2]), or the related analysis was incomplete (e.g., due to small test datasets [15]). Therefore, the algorithm proposed by Narayanan and Shmatikov in [2] was the one I considered as the state-of-the-art attack in my work, and I worked with it.

As it is hard to obtain real ground truth labeled datasets (as it was in the original experiment in [2]), in my work I used the perturbation algorithm provided by Narayanan and Shmatikov in [2]. This procedure allows to synthetically create a realistic pair of source and target graphs with adjustable strength of attacker knowledge from a single dataset (I provide more details on this in Section 3.3).

2.2 Graph Privacy Protection Methods

There are several ways for tackling re-identification attacks, such as proposing alternative social networking platforms that circumvent these attacks by nature, or proposing modifications to existing social networking services (e.g., novel algorithms for data sanitization that are applied by service providers). Finally, there is also the possibility to propose client side applications and other solutions working under the control of the user.

However, getting mainstream social platform providers to adopt modifications is not an easy task: these parties are financially interested in collecting vast amounts of (sensitive) data, and regarding what we could learn from the Snowden revelations (e.g., [3, 38]), it is reasonable to assume that relying on the service provider for protecting user privacy is not an acceptable idea. In addition, even if service providers would try to sanitize data, it has been shown that de-identification is a hard problem in general [39], and preserving graph privacy has limits if utility is also concerned [40].

There were various proposals of new service models that aimed circumventing mainstream social networking services in order to give back control to users; however, even initially highly popular and known alternatives (e.g., distributed social networks like diaspora* [7] and Safebook [41]) were not able to break into the mainstream, despite the fact they could have provided a higher level of privacy than regular social networks.

Therefore the most viable alternatives are client side solutions that do not need the consent of service providers or other parties, but can be applied to existing services. These solutions should be capable of either hiding user information or preventing large-scale re-identifiation somehow. 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 [42].

However, to the best of my knowledge, only the work of Beato et al. propose a client-side solution in [1] where a model level evaluation is also provided (beside my works in [C2, J1–J4]). In their work, they proposed the friend-in-the-middle model (FiM), where proxy-like nodes act as mediators to hide connections, successfully tackling the attack when approx. 10% [1] of nodes adopt their model. The viability of the FiM model is demonstrated on two snapshots of the Slashdot network (obtained from the SNAP collection [43]). As the friend-in-the-middle model focuses on hiding connections, it is not possible in their model to hide profile attributes, and three-party negotiation of hiding a single edge also makes the adoption of the technique a little cumbersome.

2.3 Privacy-Enhancing Identity Management

In Chaper 5 and 6 I propose the use of the privacy-enhancing identity management[1], or PIDM in short, as a solution for tackling re-identification. PIDM allows the user of a system to create multiple representations of himself, which can be either partially overlapping at some level or independent, and these are called partial identities. Partial identities can be virtual identities or profiles, that can be used for different kinds of transactions or sharing information. Identity separation is a part of PIDM, where partial identites are unlinkable to each other.

Before the discussion of applications, it must be noted that identity separation is already in use in real-world scenarios. There is a long list of authors who used pen names for several reasons[2], 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 (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[3]).

2.3.1 Brief Literature Survey of Identity Separation

Beside some early applications having functionality that resembled the concept of using partial identities [44, 45], privacy-enhancing identity management was first described by Clauß and Köhntopp in [46] as a standalone concept. This was followed by several works further developing the concept (e.g., [25, 47–49]), then the PRIME Project [50] added comprehensive details to emerge it to the framework level. Successor of PRIME, the PrimeLife Project [51] resulted in ready-to-use applications having identity management functionality.

To the best of my knowledge, we were the first in 2009 to propose the use of identity separation in social networks in [C7], where we proposed a modified social network model with a non-flat structure. The works of van den Berg and Leenes in 2010 [52,53] provided further details on identity partitioning, especially focusing on access control and division of information shared. The first public social network implementation is called Clique [54], which was developed in the PrimeLife project, and Google+[4] was the first mainstream social network adopting a feature resembling the functionality of

[1] Several alternative terms are used in the literature, from which the following ones are the most frequent: partial identities, role-based privacy, role-based access control.

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

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

[4] https://plus.google.com

identity management in 2011 [55]. The goal of this feature (namely Google Circles) is to allow proper audience selection for sharing content, which was then adopted widely in the mainstream as well.

2.3.2 Application of Identity Separation in Social Networks

From a graph privacy point of view, we can think of identity separation as a tool that simplifies the creation and management of parallel, unlinkable identities of one member. When someone adopts identity separation, that will result in multiple nodes in the observable private graph that the attacker aims to de-anonymize, each partial identity having a subset of the original contacts, but not necessarily all of the original acquaintances need to be presented. Here it also need to be assumed that the identity separation process needs to be hidden from all parties; this could be achieved if partial identities are also developed in parallel.

Compared to the friend-in-the-middle model [1], the features of identity separation facilitate hiding profile information in addition to making relationships private [25], with less cooperation (as the model in [1] required the cooperation of three nodes for hiding a single edge). Identity separation also can enable hiding user identity if the original identity of such a partial identity cannot be recovered.

In addition to the possible technical advantages, identity separation has deeper roots that motivates its use in social networks. In real life, we also classify and group our social contacts rather than simply label all of them as friends [56]. We keep track of multiple groups of people we know from different stages and roles of our lives (e.g., school, workplace, and family), and interact with them in a different fashion concerning the given context [57]. Privacy issues in social networks in general [58], and the lack of having a fine-grained access control yielded for introducing identity management into social networks.

Chapter 3

Methodology and Models

Beside describing the basic notation used in my work, this chapter mainly focuses on the methodology of simulation experiments, and modeling. I introduce the threat model in Section 3.1. In Section 3.2 I present the notation and definitions, then in Section 3.3 I discuss the core principles of data selection and perturbation. Then, in Section 3.4, I provide the settings used in experiments for simulating re-identification attacks, and finally I introduce the used identity separation models in Section 3.5.

3.1 Threat Model

In my work I use the following threat model. (The only exception is Section 6.4.2 where differences and the model are described in details.) There is an adversary, who obtains a dataset which is sanitized (i.e., without identifiers), but contains valuable, private information (right in Fig. 3.1). If there is a provider of the dataset such as a social networking service, his actions are considered to be negligible respecting the privacy of the network (e.g., using light anonymization techniques only). The goal of the adversary is to re-label the nodes in order to monetize the private information more efficiently. As the attacker uses structural information only, he acquires another social graph where node identities are known (left in Fig. 3.1). Then the attacker runs a re-identification attack that produces one-to-one mappings between the networks, revealing (probably) the truth identities of previously anonymous entities.

In my dissertation, I assume that prior the adversary obtained the sanitized dataset, users had the chance to use identity separation techniques to enhance their privacy. It is assumed that the separation procedure is done on the client side, and neither the social networking service provider (if there is any), nor other entities know the details of it (in case of cooperating users, any related information is held as a secret). The background knowledge is assumed to be a regular social network without identity separation; analysis of further settings are assigned as future work. As identity separation

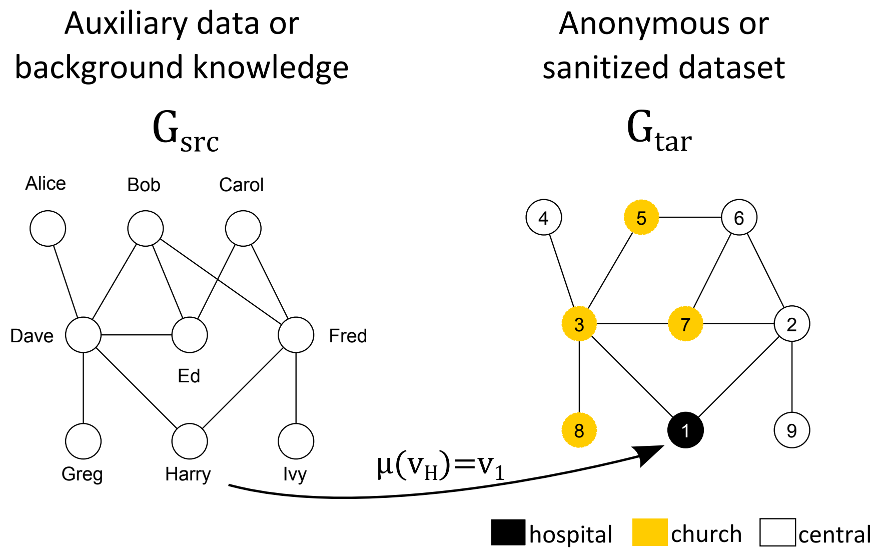


Figure 3.1: Threat model. The attacker obtains a sanitized dataset (right) that contains valuable, private information, and then runs a de-anonymization attack to recover the identities by using publicly available information acquired from other sources (left). Re-labeling is done by mapping known identities to anonmyous ones (denoted with function $\mu(\cdot)$).

damages network structure, adopting users expect that re-identification would be less successful afterwards.

3.2 Notation and Measuring Re-identification

In my work, I denote graphs as $G = (V, E)$, a structure consisting of a node and edge sets. Graphs are assumed to be undirected simple graphs – no loops and no multiple edges are allowed. I denote a vertex as $v_i \in V$, and the set of its neighbors as

$$V_i = \{\forall v_j : \exists (v_i, v_j) \in E\}. \tag{3.1}$$

Similarly to this, we can denote the vertices that are at a distance of two from v_i (or also known as friends-of-friends), as V_i^2.

Given a sanitized graph G_{tar} (target graph) to be de-anonymized by using an auxiliary data source G_{src} (where node identities are known), let $\tilde{V}_{src} \subseteq V_{src}, \tilde{V}_{tar} \subseteq V_{tar}$ denote the set of nodes mutually existing in both. Ground truth is represented by mapping $\mu_G : \tilde{V}_{src} \to \tilde{V}_{tar}$ denoting relationship between coexisting nodes. Let us denote a vertex set V as V' after having identity separation adopted (by some or all of its nodes). I denote the set of nodes before adopting identity separation as $V_{ids} \subseteq V_{tar}$, and denote $\tilde{V}_{ids} \subseteq \tilde{V}_{tar}$ the subset coexisting nodes; thus $\tilde{V}'_{ids}$ contains multiple identities of nodes from $\tilde{V}_{ids}$. Let $\lambda_G : \tilde{V}_{src} \rightrightarrows \tilde{V}'_{ids}$ denote the ground truth mappings between coexisting 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 $\mu_0 : V_{src} \to V'_{tar}$ results in a re-identification mapping denoted as $\mu : V_{src} \rightarrow V'_{tar}$.

Let denote the corresponding nodes in different networks as $v_n^{src} \in V_{src}$ and $v_n^{tar} \in V_{tar}$. *Identity separation* of user $v_n^{tar} \in V_{ids}$ is denoted and modeled as follows[1]. First, v_n^{tar} creates a total of y new partial identities which are denoted as $v_{n\backslash i} \in \tilde{V}'_{ids}$ ($i \in [1,\ldots,y]$), and then distribute edges between new identities. If there is a single partial identity that is assumed to be sensitive it is denoted as $v^{\star}_{n\backslash i}$. It is assumed that the attacker only captures the sanitized dataset after the user committed identity separation, and knows no information about the identity separation process itself. The goal of the attacker to obtain mappings of $\mu(v_n^{src}) = v_{n\backslash i}$, or to find $v^{\star}_{n\backslash i}$ in some cases where more than one of the partial identities are found.

I use two measures for assessing the extent of what the attacker could learn from μ. The *recall rate* reflects the extent of re-identification, describing success from an attacker point of view. This itself can be used due to small error rates. As identity separation is a personal information hiding tool, the quantity of information the attacker gained access to should also be concerned, which is quantified by the *disclosure rate* . This describes an overall protection efficiency from a user point of view.

Now we can describe the mode of calculation of these rates. The *recall rate* is calculated by dividing the number of correct identifications with the number of mutually existing nodes (seeds are excluded from the results). The score of a node $v^{src} \in \tilde{V}_{src}$ regarding a given re-identification mapping μ can be expressed as:

$$s(v^{src},\mu) = \begin{cases} 0 & \text{if } \nexists \mu(v^{src}) \\ 1 & \text{if } \mu(v^{src}) = \mu_G(v^{src}) \vee \mu(v^{src}) \in \lambda_G(v^{src}) \\ -1 & \text{if } \mu(v^{src}) \neq \mu_G(v^{src}) \wedge \mu(v^{src}) \notin \lambda_G(v^{src}) \end{cases} . \tag{3.2}$$

We can now quantify the *recall rate* of an attack resulting in mapping μ can be calculated as

$$R(\mu) = \sum_{\forall v^{src} \in \tilde{V}_{src}} \frac{s(v^{src},\mu) \cdot \max(0, s(v^{src},\mu))}{|\tilde{V}_{src}|}. \tag{3.3}$$

The maximum of recall is denoted as R_{max}.

The *disclosure rate* can be calculated in a similar manner. As current identity separation models are bond to structural information, the measure reflects the average percent of edges that the attacker successfully revealed (this can be extended for further types of information in other experiments, e.g., sensitive profile attributes). The disclosed information can be quantified for an individual node $v_n^{tar} \in \tilde{V}_{ids}$ as

[1] While this is acceptable for modeling, in real life partial identities should be developed in parallel in order to retain information on the identity separation process

$$d(v_n^{tar}, \mu) = \begin{cases} \frac{deg(v_{n\setminus i})}{deg(v_n^{tar})} & \text{if } \exists \mu(v_n^{src}) = v_{n\setminus i} \wedge v_{n\setminus i} \in \lambda_G(v_n^{src}) \\ 0 & \text{otherwise} \end{cases} . \tag{3.4}$$

By using this function we can now define the disclosure rate of the attacker over the nodes applying identity separation w.r.t. mapping μ as

$$D(\mu) = \sum_{\forall v_n^{tar} \in \tilde{V}_{ids}} \frac{d(v_n^{tar}, \mu)}{|\tilde{V}_{ids}|}. \tag{3.5}$$

The *re-identification rate of a node v* in a series of experiments ν is considered in some cases, which is calculated as

$$S(v) = \sum_{\forall \mu \in \nu} s(v, \mu), \tag{3.6}$$

where $s(v, \mu)$ can theoretically take arbitrary values in the series of ν. However, as the Nar09 algorithm is quite deterministic (I show this in Section 3.4), negative and positive values of $s(v, \mu)$ are typical to occur for the same node in a series of experiments.

3.3 Data Sources and Perturbation

Data selection for simulational evaluation should be done carefully, at least for two reasons. The class of structural attacks discussed are capable of achieving large-scale re-identifications rates even in networks significantly larger than tens of thousands of nodes. Therefore, evaluation of protective measures and novel attack algorithms can be executed in a plausible way when large networks are involved. Second, network structure can bias results, therefore evaluation need to be executed on multiple datasets obtained from different sources. As I highlighted it in Section 2.1, conforming these rules is not prevalent in the literature, unfortunately; however, data selection in my work was done with having these considerations in mind.

During most of the experiments (differences are marked) I used multiple datasets with different characteristics in order to avoid biases caused by the structure, and these were large networks where brute-force attacks are practically not feasible. For keeping measurements realistic, datasets were obtained from real networks. Unless otherwise stated these were downloaded from the SNAP repository [43], where details description and statistics are also available. The first network I used was the Slashdot network crawled in 2009 (82,168 nodes, 504,230 edges). Slashdot is a technology-related news site where users can specify which others users they know. I also worked with the Epinions network crawled in 2002 (75,879 nodes, 405,740 edges). The Epinions website provides consumer reviews, and the extracted social graph is based on the who-trust-whom relations. The third dataset is a subgraph exported from the LiveJournal network crawled in 2010 (at our dept.; consisting of 66,752 nodes, 619,512 edges). LiveJournal is a blogging service, and its social network graph is based on how blog owners declare who their friends are within the service. In some cases I also used two smaller datasets for comparison, one is from the LiveJournal crawl denoted as LJ10k (10,056 nodes, 231,416 edges), and the other is the Wikivote dataset (7,115 nodes, 100,762 edges) also obtained from the SNAP collection. The Wikivote dataset contains who-voted-for-who information during the promotion of users to obtain admin privileges on the website of Wikipedia.

For generating test data, first a background knowledge (G_{src}) and a target graph (G_{tar}) is derived 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}, I used the perturbation strategy proposed by Narayanan and Shmatikov [2], which produces realistic test data. Their algorithm works as follows: it derives G_{src}, G_{tar} with the desired fraction of overlapping nodes (α_v) from the source dataset simply by splitting the original vertex set. All original edges are preserved. Then edges are deleted independently from these copies to achieve an edge overlap α_e. Ground truth μ_G can be easily calculated during the procedure. These parameters model the strength of the attacker, as higher overlaps stand for stronger attackers having relevant background knowledge.

After running several measurements with different settings for α_v, α_e, I used $\alpha_v = 0.5$, $\alpha_e = 0.75$ in the experiments of the dissertation, unless otherwise stated. This setting is a good trade-off at which a significant level of uncertainty is present in the data (thus noisy and life-like), but it is still possible to identify a large ratio of the co-existing nodes[2]. For comparison, in Table 3.1, I provided recall rates for the Nar09 algorithm for different settings for the main datasets.

Next, I modeled identity separation on the target graph. Then nodes are split and their edges are sorted according to the settings of the currently used identity separation model (for node sampling methods and models see Section 3.5). By recording these operation, we can extend the ground truth mapping μ_G with λ_G.

The setting I used my work is that the sanitized dataset have identity separation, while the background knowledge is a regular social network. I used this setting to measure the viability of identity separation against re-identification. However, in future work additional settings should be also measured. When the attacker obtains background knowledge that contains identity separated users, is not automatically means a

[2] Note: choosing different values would not affect results in general, but only determine the maximum recall rate achieved without having identity separation introduced to the simulations.

		Slashdot				Epinions				LJ66k			
		α_e				α_e				α_e			
		0.25	0.5	0.75	1.0	0.25	0.5	0.75	1.0	0.25	0.5	0.75	1.0
α_v	0.25	0.6	2.6	11.7	19.8	1.1	5.1	10.9	14.8	0.8	6.3	19.5	27.6
	0.5	0.4	19.8	36.5	47.5	0.9	17.2	25.9	32.6	0.6	24.8	35.7	54.4
	0.75	0.3	33.6	50.7	60.4	0.6	25.3	36.1	44.4	0.7	33.9	57.9	78.7
	1.0	0.3	30.1	58.9	68.3	0.4	31.2	43.2	52.5	1.5	37.7	75.2	88.5

Table 3.1: Recall rates were proportional to the overlap between G_{src} and G_{tar}: the less perturbation is used (resulting higher overlaps) the higher recall rates are. The table clearly shows that the Nar09 is more sensitive to the proportion of missing edges. Note: it is not possible to achieve 100% recall as there can be structurally equivalent nodes in the datasets, and low degree nodes are harder to be re-identified in general (see Fig. 3.4).

full privacy breach. As the identity separation process is assumed to be done secretly, the attacker could use this background knowledge to reveal hidden attributes in the identity separated anonymous network, but this information could not be linked to the real identity of the user. Here, future work should focus on finding appropriate strategies for using identity separation in order to prevent such information leaks.

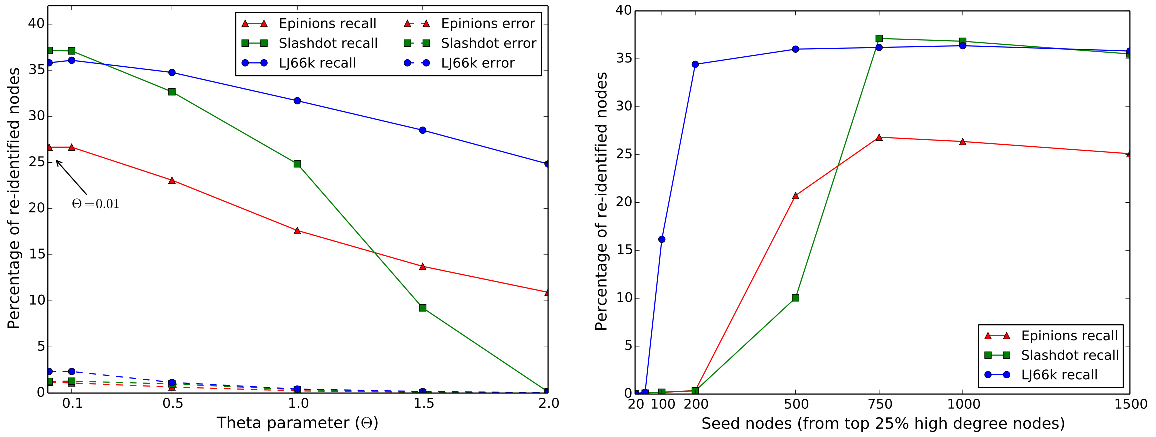
3.4 Working with the Nar09 Algorithm

In this Section I provide the details how I used the Nar09 algorithm in simulation experiments. In general, I used Nar09 by default for simulations; however, in some well marked cases I have additionally provided results for comparison with the Grasshopper attack. In these cases, except using a lower number of seed nodes (100 seeds of the same type), I used the same parameters as of Nar09 (e.g., for Θ), unless otherwise stated.

3.4.1 Settings of the Algorithm

In each experiment I created two perturbations, then run simulations two times on both with another seed node set (different settings are noted). As only minor deviations were observed in results, usually less than a percent, I found this setting to be suitable. I have also compared the directed and undirected versions of Nar09, but only negligibly small differences occurred, thus for the sake of simplicity I worked with the undirected version (pseudo code is provided in Appendix A.1).

Probably the most important parameter of Nar09 is Θ, controlling the ratio of true positives (recall rate) and false positives (error rate). The lower Θ is the less accurate mappings the algorithm will accept. As I measured fairly low error rates even for small values of Θ, I have chosen to work with $\Theta = 0.01$. In the majority of experiments the ground truth error rate (later referred as the error rate) stayed typically around



(a) Varying the Θ parameter on perturbed networks (with `random.25`).

(b) Phase transition property illustrated for the `random.25` seeding method in different networks.

Figure 3.2: Measuring basic properties of the Nar09 algorithm.

a few percents (e.g., Fig. 3.2a). The overall error was around 5% without identity separation, and decreased significantly when identity separation was applied.

The seeding method and size is another important property of Nar09. To the best of my knowledge, we were the first to elaborate the importance and effects of this property in details, also resulting in providing guidelines in [C1]. For example, the algorithm has a phase transition property [2, 59], e.g., resulting in network structure biases on the minimum number of seeds respecting when the given seeding method could result in large-scale propagation. An example is provided in Fig. 3.2b explaining this property, and I discuses the relevance of further properties in Section 4.2.

For initializing Nar09, in general, I applied random seed selection of high degree nodes selected from the top 25% (later this is denoted as `random.25`). However, as in my work I also characterize the importance of seeding in Section 4.2, for evaluation I also use some other seeding methods as well (deviance is clearly stated).

3.4.2 Further Properties Determining Results

I have characterized properties of Nar09 algorithm that were not presented explicitly in the literature in details. One finding is related to low error rates, namely that the distribution of $S(v)$ is quite unique. For the majority of nodes, it is quite deterministic (regardless of the current instance of seed nodes) which nodes the algorithm can and which it cannot find. For all the experiments enlisted in Table 3.1, 84.63% of nodes had $S(v) = 0$ or $S(v) = 10$, and for cases with higher recall 93.83% of all nodes fall into this category; for details, see Fig. 3.3.

Node degree is important for the propagation phase of the Nar09 algorithm as it fingerprints nodes by their connections: the higher node degree is, with higher confi-

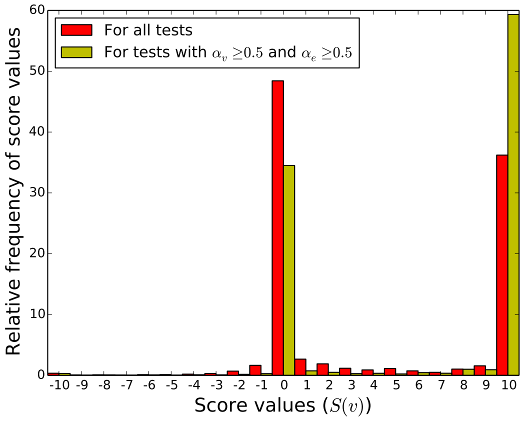


Figure 3.3: Distributions of node re-identification score values ($S(v)$) over the test results presented in Table 3.1.

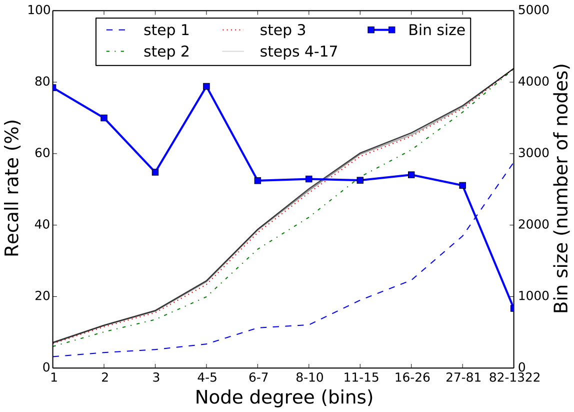


Figure 3.4: Degree distribution of correctly re-identified nodes in subsequent propagation steps on the LJ66k dataset (seeds excluded). Re-identification runs while there is convergence.

dence it can be compared with others. In Fig. 3.4 I displayed the subsequent steps of a run of the Nar09 algorithm on the LJ66k network with no identity separation. As it is shown, almost maximum propagation is achieved in the first two steps, followed by a rather slow convergence. However, the figure also shows that the algorithm performs significantly better in re-identifying high degree nodes, i.e., while almost all high degree nodes were re-identified, the recall rate was under 20% for nodes with $\deg(v) \leq 3$.

3.5 Modeling Identity Separation

For considering identity separation for simulations, basically there are two settings: non-cooperative and cooperative, representing cases when users make decisions on adopting identity separation on their own or collaborate in some ways. Regarding simulations, in case of the non-cooperative setting, node are uniformly sampled from the target graph (where $\deg(v) \geq 2$), but in case of cooperation nodes are sampled accordingly to the considered collaboration scheme.

There are several ways how cooperative identity separation can be realized; in my dissertation I consider cooperation schemes that are organized either locally or accordingly to the global importance of the selected nodes. In local cooperation neighboring nodes use identity separation together, independently in randomly selected parts of the network, while in the global case nodes are selected according to a global measure of importance. However, it must be noted that further schemes could be considered, e.g., finding cuts for separating the network.

3.5.1 Analyzing Real-life Datasets

It would be desirable base identity separation models on characteristics observed on datasets obtained from real-life sources. In order to do this, these datasets should include behavioral patterns on the number of identities users create, how users sort edges between these identities, and how privacy is preserved in these services. The latter property eventually results in missing edges and partial identities, which is not visible in most released datasets. Therefore, to my knowledge, there are no datasets presenting this property and there are no trivial ways of crawling one (yet). For the two former properties, there are datasets on ego networks describing these ones.

Ego networks have similar functionality to identity separation, as they represent how users create partial identities to manage their social connections [56]. There are related datasets available from Google+, Twitter and Facebook on the SNAP repository [43]. By analyzing this data, I found that the number of circles has a power-law distribution, for instance in the Twitter dataset I measured $\alpha = 2.31$ (933 ego networks, $x_{min} = 2$, $x_{max} = 18$). Many users did not duplicate any of their connections (44.6%), and only a fragment of them had more than twice as many connections in their circles compared to the number of their unique acquaintances (6.07%). While it is not possible to draw strong conclusions from these observations (as there are no patterns on hiding information), I believe these indicate the possible nature of identity separation.

3.5.2 Pseudonyms as the Basis of Private Identities

Privacy-enhancing identity management in social networks need to be based on the use of pseudonyms[3], which are the explicit identifiers assigned to the partial identity of the user (i.e., pseudo-names). These named identities then can be used within several contexts, and changed if needed – the given methods were elaborated in great details in the PRIME Project [50].

In order to achieve identity separation with PIDM, these pseudonyms should be unlinkable by trivial means, e.g, the user's pseudonyms must not disclose that they belong to the same user or to the real identity of a user [60]. At the model level, a social network supporting identity separation should allow the following levels of identification:

Total identification Within the highest level of identification the user can be identified with his real identity. For instance, this is the expected case for Facebook, where only valid identities are permitted for registration[4].

[3] However, an implementation should also consider that individual attributes, behavioral patterns could also be used as quasi-identifiers, and such risks need to be mitigated.

[4] Without any verification.

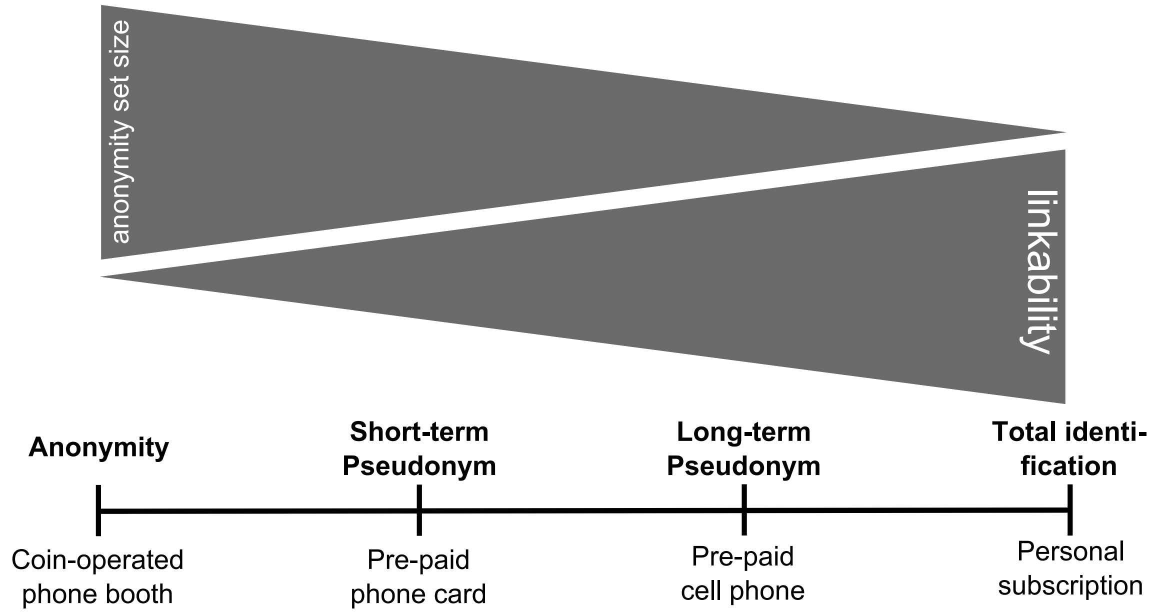


Figure 3.5: Different levels of anonymity explained with phone identifier example. The longer an identifier is used (in more situations, for contacting more entities), the more individually identifying it is.

Pseudonymous identification This level of identification is bounded to a pseudonym. These identifiers can have several properties that can limit their uses. There are different types of pseudonyms such as person, role, relationship, and transaction pseudonyms, which can have different lifetimes and provide different levels of privacy given by the linkability across contexts [61] (e.g., linkability of each pseudonym to past activities). Within the context of identity separation, personal pseudonyms have the weakest level of privacy, as these are globally unique identifiers (sometimes even not changable) linking user actions on the global scale. Unlinkable pseudonyms have a stronger level of privacy, where the user may separate his identities by the means of multiple unlinkable pseudonymous identifiers.

Anonymity The strongest level of privacy is provided by anonymity, which allows actions without identifiers. A similar level of privacy can be achieved when pseudonyms are used only per transaction (i.e., transaction pseudonyms).

Depending on how long a pseudonym is used, and also considering the number of actions it was used with, and the number of contexts where it appeared, each type of pseudonym provide a different level of privacy. The use of pseudonyms is illustrated in Fig. 3.5 with different levels of phone-related identifiers.

Using identity identity separation in such a way will be reflected on the network structure. Managed, but not separated identities (i.e., those that use linkable pseudonymous identifiers) should appear as a single node, unlinkable separated partial identities will appear as separate vertices. As the current model allows users to hide identities and edges, these should not appear in the network topology at all.

3.5.3 General Model and Formalization

As described earlier, in order to commit identity separation, v_n^{tar} creates a total of y new partial identities which are denoted as $v_{n\backslash i} \in \tilde{V}'_{ids}$ ($i \in [1, \ldots, y]$), and then sorts edges between new identities according to a given distribution $p_1, p_2, \ldots, p_y$. The number of new identities (y) is modeled with a random variable Y. The distribution of the edge sorting is $P(X_1 = x_1, \ldots, X_y = x_y)$, where X_i is a random variable describing the degree of $v_{n\backslash i}$. At this point, there is no presumed distribution for Y, and the distribution for X_i is defined later in coherence with the chosen sub-model.

The model in general is based on the following assertions about the structure of the network before and after applying identity separation. These assertions are assumed to be true in all sub-models.

General assertion 1 A new identity can have even zero of the original contacts (i.e., the user chooses absolute privacy for that identity).

General assertion 2 A user v_n^{tar} may create a maximum of $\deg(v_n^{tar})$ new identities. While it is not possible limit the number of new identities in theory, it would not match with the user's expected behavior, and thus it is an acceptable rational limitation.

General assertion 3 A user may create even 0 new identities (i.e., absolute privacy for the user).

General assertion 4 No new additional edges are created during identity separation (e.g., for the purpose of deception). This simplifies the behavioral model, but it might be a desirable functionality to be introduced as future work; this could add useful noise to deceive the attacker.

General assertion 5 Edges are not sorted independently. This is a rational consideration, since all new identities belong to the same user, who sorts the edges in an intelligent way (or the software agent supporting the PIDM for the user).

Simulation application of the models build on these assertions, however, these will be even more important in the formal analysis of identity separation.

3.5.4 Submodels

Dependent on the chosen user behavior, there are further aspects to be considered in the submodels (with $0 \leq x_i \leq n$):

- Can different identities of the same user have overlapping neighborhood (i.e., duplicated edges)? Overlapping allows the overall number of connections to increase, formally, $\exists P(X_1 = x_1, \ldots, X_y = x_y) > 0$, that $\sum x_i > n$.
- Is edge anonymization permitted? Deleting edges allows the overall number of connections to decrease, as $\exists P(X_1 = x_1, \ldots, X_y = x_y) > 0$, that $\sum x_i < n$.

Based on these aspects, new submodels can be introduced that are summarized in Table 3.2. The names of the submodels require some explanation. I have named the model with no edge anonymization, and no overlaps the basic model, since this allows the least functionality for the user (only identity separation itself). Conversely, the realistic model is just the opposite: it implies the fewest limitations on user actions. Users of a social network would likely use the functionality of this model (e.g., duplication of some edges and the deletion of others); hence the notation realistic.

	Overlap	**No overlap**
Edge deletion	Realistic model	Best model
No edge deletion	Worst model	Basic model

Table 3.2: Submodels for identity separation with considering possible functionality.

Besides, a worst and a best model also exist, which are also named from the user-centered point of view. The best model allows a user to only decrease the number of his contacts, and therefore causing more information loss to the attacker, therefore preserving more privacy. The worst model is the opposite: it only allows creating multiple connections between identities and acquaintances, therefore making "backups" of structural information, and helping identification. According to Table 3.2, further assertions can be characterized for each model including limitations to edge deletion, duplication functionality (these are detailed later).

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). Where this model is used, edges are sorted with uniform probability between new identities. The realistic model is used to describe real-life behavior (where both edge deletion and multiplication allowed), and the best model allows describing privacy oriented user behavior (no edge duplication, but deletion allowed). While I explicitly omitted the worst model (edge duplication only), I must note that there are similar experiments in my dissertation: the behavior patterns from the Twitter network are the closest to this model, as these patterns contain duplication but no deletion (see Section 5.3). I provide further details on how the sub-models are used with the analysis.

3.5.5 Probabilistic Basis for the Models

In the basic model, the user v_n^{tar} sorts $N = \deg(v_n^{tar})$ edges among y identities. The multinomial distribution is a natural choice for describing such a case (also because of General assertion 5), since it describes N trials when the outcomes can be sorted into one of y groups. Additionally, group probabilities can be adjusted, and therefore this model allows fine-tuning the distribution in a way for describing user behavior in the desired way. Multinomial distribution is used as

$$P(X_1 = x_1, \ldots, X_y = x_y) \sim Mu(N, p_1, \ldots, p_y), \tag{3.7}$$

where $\sum p_i = 1$.

The realistic model is more flexible than the basic model, as it allows for edge deletion and also duplication. Due to this flexibility, there is no predetermined distribution that could be used for conclusive analysis. To be rather realistic (and harmonize observations presented in Section 3.5.1), the distribution should reflect that the most likely case is that the number of all contacts after identity separation is similar to degree of the node before, i.e., a few deletions and duplications are likely to happen, but major deviations are not.

For analytic approaches (Section 5.1), I use the sum of binomial distributions as depicted in Fig. 3.6 to capture the expected behavior of the realistic model. In simulation experiments, the realistic and best models are both used to test the propagation of Nar09 against edge perturbation added by identity separation. For the experiments edge sorting probabilities are calculated according to multivariate normal distribution, denoted as

$$P(X_1 = x_1, \ldots, X_y = x_y) \sim \mathcal{N}_y(\boldsymbol{\eta}, \boldsymbol{\Sigma}), \tag{3.8}$$

where y denotes the current number of identities. The value of $\boldsymbol{\eta}$ was set to $(y)^{-1}$ and I configured $\boldsymbol{\Sigma}$ to maintain the overall number of edges during the process, allowing only a little number of added or removed edges. In the best model, if the sum of new edges exceeded the original degree, the distribution was simply recalculated.

Based on this, a new edge distribution is randomly selected as $(x_1, \ldots, x_y)$. The *realistic model with minimal deletion*, in which every edge is assigned to one identity, and if there is still ample space left for edges, new copies of edges are assigned randomly. In this setting edges are not deleted unless it is necessary. In the setting of the *realistic model with random deletion* new identities take a portion of edges proportional to

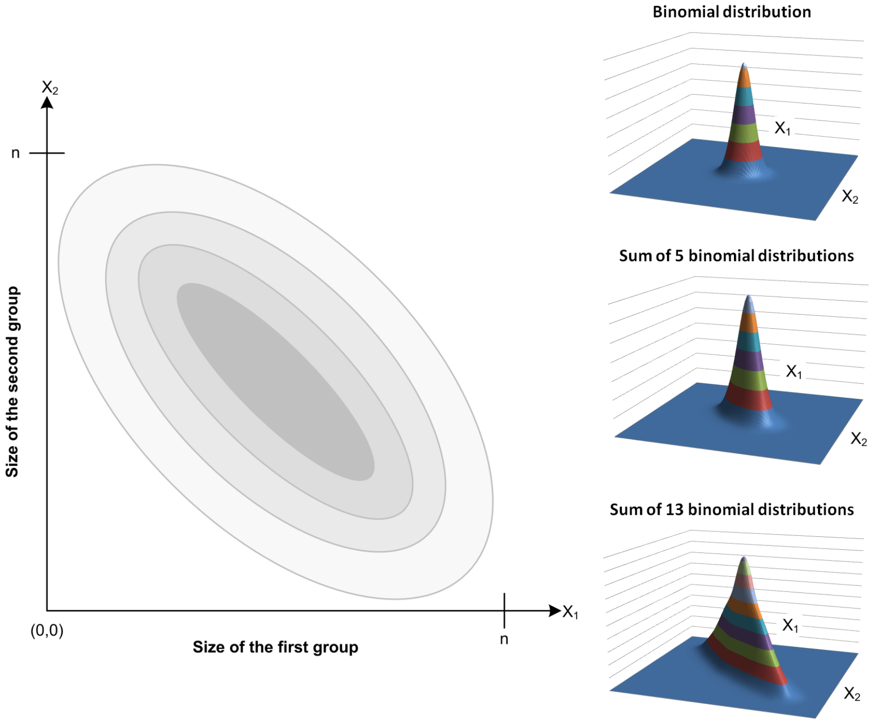


Figure 3.6: Concept for the distribution of realistic models with $Y = 2$, and some examples. On the left part of the figure, the darker areas have higher probabilities (these values are outstanding on the right part).

$(x_1, \ldots, x_y)$, leading to delete unassigned edges proportionally to $\prod(1 - \frac{x_i}{\deg(v_n^{tar})})$. I also included a setting with the best model called the *best model with random deletion*. However, it should be emphasized that none of these settings capture aggressive edge deletion, but it might be interesting to investigate such settings in the future.

Chapter 4

Analysis of Structural De-anonymization Attacks

'Friends don't spy; true friendship is about privacy, too.' (Stephen King)

In Section 3.4 I discussed several parameters of the Nar09 attack, including ones that were not mentioned or detailed in the literature before. In this chapter I provide the analysis of two novel findings related to these types of attacks that bear greater importance. In Section 4.1 I propose a class of anonymity measures that can be used within the current context, and then evaluate instances of this anonymity class for the Nar09 and Grasshopper attacks. Then in Section 4.2 I argue the importance of seeding, and I show for the Nar09 algorithm how initialization effects the overall performance of the algorithm.

4.1 Measuring Structural Anonymity

In cases when structural fingerprint of a node is considered globally, it is trivial (but not necessarily effective) to measure the anonymity level of a node: the level of anonymity is proportional to the number of nodes with the same fingerprint, i.e., number of nodes being in the same anonymity set. However, in case of attacks like Nar09 nodes are compared locally, and therefore anonymity sets cannot be considered in such sense, and also cannot be calculated explicitly by trivial means. Furthermore, without possessing explicitly the background knowledge of the attacker, anonymity can be only estimated.

However, local anonymity measures are useful from a privacy-oriented point of view, as these can express the node's resistance level against local re-identification techniques, but beneficial in other ways as well. These can help users to decide how to apply privacy-enhancing software to strengthen their privacy status, more accurately than simply considering their global structural re-identifiability. These measures can support data providers and attackers to make (lower) estimates of the possible success of attacks. In my work I focus on measuring the level of anonymity for nodes, and leave the detailed analysis on how measures over the entire network could be used as future work.

Here I propose a general concept for measuring anonymity, and provide the analysis of two main approaches that could be used for Nar09. In the first case I provide non-trivial anonymity measures based on how the propagation phase of Nar09 and similar attacks work, then I show that degree can also serve as a simple anonymity measure. Finally I show that the prior measures are significantly better in several cases than node degree, and I also discuss when.

4.1.1 Local Topological Anonymity

Large-scale structural re-identification attacks compare nodes against their 2-neighborhoods in their local re-identification phase, therefore, the more similar a node is to its neighborhood, the lower chance it has for being re-identified. This property need to be captured by anonymity measures. Therefore, we can now introduce Local Topological Anonymity (LTA), which is a measure that predicts the level of anonymity against re-identification attacks based on local comparison of nodes, such as Nar09.

There can be numerous variants for node fingerprints in attack algorithms. Nar09 simply compares the sets of neighbors of nodes (of G_{src}) to the neighbors of their friends-of-friends (in G_{tar}). While in other attacks node neighborhood may be inspected more deeply at the expense of larger node fingerprints and increased run-time. Therefore, the concept of LTA need to be easily adoptable for these cases. Based on this, we can now define LTA:

Definition 1. *A Local Topological Anonymity measure is a function, denoted as $LTA(\cdot)$, which represents the hiding ability of a node in a social network graph against attacks considering solely the structural properties of the node limited to its d-neighborhood*[1]*.*

Then LTA variants tailored for specific attacks or based on specific principles can also be defined:

Definition 2. *A Local Topological Anonymity measure variant α is a function, denoted as $LTA_\alpha(\cdot)$, which is an LTA measure that is based on the node fingerprint function $f_\alpha(\cdot)$ representing the structural fingerprint of a node in a social network graph.*

[1] In my work I used $d = 2$ as using larger distances are not feasible due to small network diameter.

As LTA is defined to be calculated on a limited neighborhood size (by parameter d), it should give a fast way to calculate an aposteriori anonymity measure regardless of the background knowledge of the attacker. By having smaller values for d, it requires inputs that can be reasonably assumed to be available for users in most services: the neighbors and neighbors of neighbors of nodes.

Node Fingerprint and Similarity

However, in order to propose specific measures, node fingerprint functions need to be characterized first. In case of the Nar09 algorithm this is a simple task: nodes within the distance of $d = 2$ are compared by checking their neighbors. Thus the node fingerprint function gives the set of neighbors as a result:

$$f_{\mathrm{Nar09}}(v_i) = V_i = \{\forall v_j : \exists (v_i, v_j) \in E\} \tag{4.1}$$

The next question is how the comparison should be done. While Nar09 compares nodes in a way of resembling cosine similarity, there are yet a great variety of other similarity measures that could be check as alternatives (especially when LTA is tailored for another algorithm). In case of the evaluation of similarity measures, we need to consider also that good candidates need to meet the following properties:

- *Symmetric*: can be compared both directions (background knowledge and sanitized data)
- *Normalized and positive values*: independent of network size, can be easily compared.
- *Fast to calculate and non-recursive.* Ease (and speed) of calculation and requires little knowledge.

Although I initially considered a great variety of measures (e.g., Jaccard [62], IDF [63], Adamic/Adar [64], Pearson [32], Simrank [65], topological overlap [66], LogOdds [67]), and also run comparative measurements with them, most did not meet the previously enlisted properties or had very differing results compared to cosine similarity. Fortunately, there is other comparative evaluation in the literature that helped in the selection.

Spertus et al. [67] conducted a comparative evaluation of similarity measures that could be used for recommendation. In their experiment, similarity of users was calculated based on their community subscriptions regarded as sets, which is quite similar to our current case, where a node can be fingerprinted as a set of its neighbors, and two nodes can be compared accordingly. Finally, they found that cosine similarity is the best measure for such purposes. Due to these reasons, and the fact that cosine similarity is both used in [2, 14], I chose cosine similarity for the proposed LTA variants. This can be written as

$$CosSim(v_i, v_j) = \frac{|V_i \cap V_j|}{\sqrt{|V_i| \cdot |V_j|}}, \tag{4.2}$$

where v_i, v_j represents nodes, and V_i, V_j represents the sets of their neighbors respectively.

Proposal of LTA Variants

Based on the previous discussion, I propose three variants based on $CosSim(\cdot)$ (which can be replaced if needed). LTA_A specifies the average similarity of a node compared to others in its 2-neighborhood (i.e., friends-of-friends). LTA_B uses a different normalization scheme than LTA_A, i.e., the degree of the node, but at least two. Here, the intuition on the normalization is to penalize higher degree nodes, as they can be re-identified more easily. LTA_C further divides LTA_A with the standard deviation of the difference in degree values between v_i and members of V_i^2 (i.e., capturing the divergence of the context), which is the set of the neighbors within two hops.

These variants can be written as follows:

$$\text{LTA}_A(v_i) = \sum_{\forall v_k \in V_i^2} \frac{CosSim(v_i, v_k)}{|V_i^2|}, \tag{4.3}$$

$$\text{LTA}_B(v_i) = \sum_{\forall v_k \in V_i^2} \frac{CosSim(v_i, v_k)}{max(|V_i|, 2)}, \tag{4.4}$$

$$\text{LTA}_C(v_i) = \sum_{\forall v_k \in V_i^2} \frac{CosSim(v_i, v_k)}{|V_i^2| \cdot max(\sigma_{deg}(\Delta V_i^2), 1)}. \tag{4.5}$$

Within these cases LTA measures are expected to indicate level of identification as: the lower the LTA value is, the higher the chances are that the node will be re-identified.

As I showed in Fig. 3.4 in Section 3.4, node degree is also an important property regarding re-identification rates; for example, less than 20% of nodes with $\deg(v) \leq 3$ were correctly re-identified, while this was around 80% for high degree nodes. Therefore I additionally propose to evaluate node degree as an anonymity measure:

$$\text{LTA}_{deg}(v_i) = \deg(v_i). \tag{4.6}$$

In case of $\text{LTA}_{deg}(v_i)$ assessment of node anonymity works differently compared to the other measures, as one can expect: that the higher the node degree is, the higher

Variant	α_v	α_e	Clique overlap
WV-1000vA	0.5	0.60	117
WV-1000vB	0.5	0.75	469
WV-1000vC	0.5	0.90	1,403
WV-1000vD	0.4	0.75	115
WV-1000vE	0.6	0.75	1,264
EP-1000vA	0.4	0.45	1,536
EP-1000vB	0.4	0.60	2,093
EP-1000vC	0.4	0.75	12,271
EP-1000vD	0.5	0.60	10,495
EP-1000vE	0.6	0.60	27,227
SD-1000	0.5	0.85	1,844
LJ-1000	0.5	0.90	2,085
LJ-10k	0.4	0.60	2,422
WV-Full	0.5	0.60	4,699

Table 4.1: Initially, for the evaluation of LTA measures I used these datasets with the provided perturbation parameters. The size of each dataset is included in its name, ranging from a thousand nodes up to ten thousand. Ground truth strength between G_{src} and G_{tar} is provided in the number of overlapping 4-cliques; the more the better for the attack.

the chance is that it can be de-anonymized. For the other measures high the anonymity values indicates lower chances of re-identification.

4.1.2 Evaluation of Measures

These measures can be evaluated by measuring the correlation between the anonymity values and recall rates. Throughout this section, 10 rounds of simulation is used in the measurements. In case of variants $\text{LTA}_A, \text{LTA}_B, \text{LTA}_C$ anonymity values should rank recall rates in a decreasing order, and in an increasing order in case of LTA_{deg}.

Initial Evaluation

For initially testing the concept of LTA, I carried out some preliminary experiments on smaller and mid-sized networks (up to 10k nodes) with the variants of $\text{LTA}_A, \text{LTA}_B, \text{LTA}_C$. The main data sources to create these small datasets were the Epinions, the Slashdot, the LJ66k networks including the Wikipedia vote and the LJ10k datasets also. Perturbation parameters of datasets created for the initial testing is shown on Table 4.1.

Simulation experiments on datasets shown in Table 4.1 were initialized with random 4-cliques. Due to the small size of these network, propagation here appeared to be more sensitive to seeding than in large networks. Thus 10 rounds of simulation turned out to be essential to give a good average of re-identification rates. In each round, a network

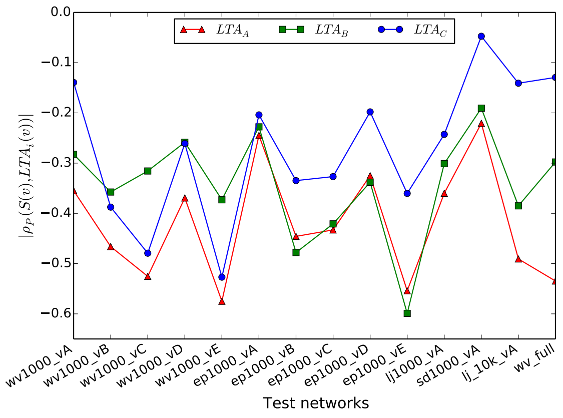


Figure 4.1: Pearson correlation coefficient values for different LTA measures in small and mid-sized networks. (dataset descriptions provided in Table 4.1)

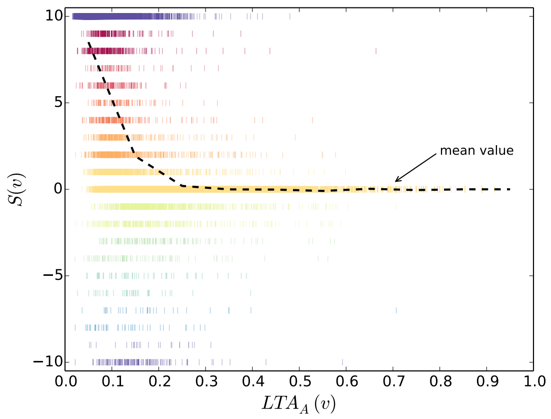


Figure 4.2: $S(v)$ ordered by LTA$_A$ scores (LJ66k). This visualization shows that LTA$_A$ is a promising measure of importance.

dependent number of coexistent 4-cliques are selected randomly (typically $7-10$). To find a clique, first an arbitrary node is selected, and then a coexistent clique is chosen from its neighborhood.

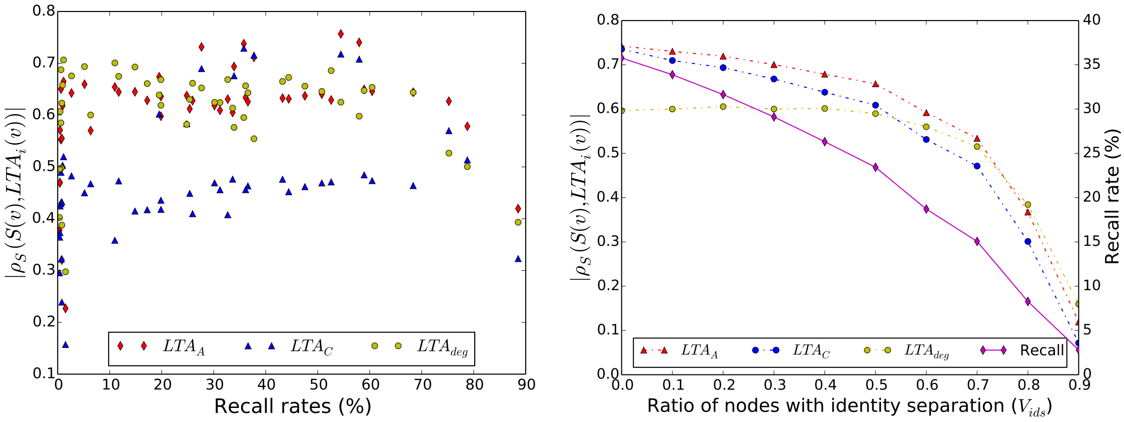
The evaluation here was done by using Pearson correlation (denoted as ρ_P) [32], where we could expect negative correlation values if measures is correct. Simulations results are depicted in Fig. 4.1. In general, all measures proved to be at least somewhat effective as all correlation values were significantly deviating from 0. Average correlation values were $-0.421, -0.344, -0.269$ for each measure respectively. In mid-sized network clearly LTA$_A$ had the best results where correlation values were between -0.5 and -0.6.

Evaluation of Measures in Large Networks

In the following experiments I measured correlation between node re-identification rates ($S(v)$) and anonymity measure values (LTA$_i(v)$) in two test sets. In the first set, I created 32 perturbations for the Slashdot, Epinions and LJ66k networks (with 16 different perturbation settings). Table 3.1 contains the recall rates of the measurements by running the attack with similar parameters as before. In the second experiment, I run identity separation with the basic ($Y = 2$) and the best models ($Y = 5$, random edge deletion) and measured correlation values afterwards.

For the correlation measurement here I used the Spearman's rank correlation (denoted as ρ_S) [68] instead of Pearson correlation, as it is more important to see if an LTA metric correctly orders nodes in a decreasing or increasing order according to $S(v)$, but the exact difference between rankings is not important. An example for the ranking is shown in Fig. 4.2. The mean value clarifies the trend that the majority of results follow.

The results of the test sets are shown in Fig. 4.3. As both correlation values closer



(a) Comparison of LTA variants with different perturbation settings, and their relation to recall.

(b) Correlation of LTA variants with identity separation in the LJ66k dataset (basic model, $Y = 2$).

Figure 4.3: Results of the first set of experiments depicted on (a) with different perturbation settings (see Table 3.1), and results of the second set of experiments plotted on (b) with different ratio of users applying identity separation. While LTA$_A$ and LTA$_{deg}$ both have the most competitive correlation values as shown in (a), in some cases LTA$_A$ is clearly the best choice as shown in (b).

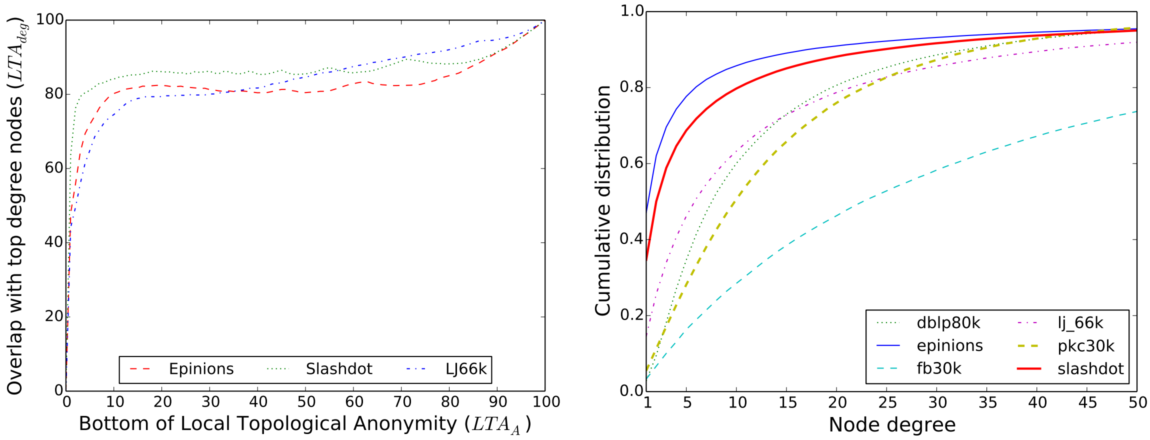
to 1.0 (for LTA$_{deg}$) and to -1.0 (other measures) are considered to be appropriate, I displayed the absolute value of the correlations. This could be done as correlation values were consequently positive or negative for each measure.

While clearly LTA$_A$ and LTA$_{deg}$ stand out as the most competitive measures in Fig. 4.3a, results in Fig. 4.3b shows a case where LTA$_A$ provides significantly better results. It can be further observed that correlation values are constant-like ($0.3 \leq \rho_S \leq 0.8$) when recall rates achieve a fair level (e.g., $5\% \leq R(\mu)$). One should also note that I omitted LTA$_B$ from these figures as it had significantly worse correlation values compared to the others, and results were almost randomly scattered around zero correlation.

When recall rates are low, only a handful of nodes are re-identified that are connected to seeds. These can have a wide range of LTA values, and this issue should account for the randomness on the lower end of recall rates (for all measures). On the higher end, the drop in correlation is caused by the fact that the majority of nodes re-identified with all kinds of LTA values; however, measured correlation values are still satisfactory as being around 0.3-0.4.

Comparison of Degree versus LTA

While comparing the two most competitive measures (which seem to have a rather high proportion of overlap as shown in Fig. 4.4a), I found that it is not the perturbation setting that seem to differ for the correlation values, but the network structure. LTA$_{deg}$ has better results in Slashdot, Epinions, while LTA$_A$ proved to have better correlation values in LJ66k. Thus I calculated the differences on the first test set as



(a) Overlap of top degree nodes and bottom LTA$_A$ (b) Degree distribution in the low degree range

Figure 4.4: Important structural properties we considered in our experiments.

$|\rho_S(S(v), LTA_A(v))| - |\rho_S(S(v), deg(v))|$, and plotted results in Fig. 4.5a, now indicating the network as well. In cases with $5\% \leq R(\mu)$ the LTA$_A$ measure is better in 13 cases with the average difference of 0.0821, while the LTA$_{deg}$ is better in 22 cases with the average difference 0.0270.

I compared the structure of the differing networks, and found that in Slashdot and Epinions the vast majority of nodes have very low degree values, e.g., the ratio of nodes with $deg(v) \leq 3$ is 58.8% and 69.6% respectively. While the degree distribution of LJ66k seem to be more balanced, and the ratio of such low degree nodes is just 33.9%. The hypotheses for interpreting the difference was that in networks with a degree distribution similar to LJ66k, LTA$_A$ captures the difference between nodes more precisely than node degree. Therefore, for verification, I aimed to compare results in additional networks that have similar degree distribution to LJ66k.

In order to do this, additional datasets were downloaded from the SNAP [43] and the Koblenz [69] collections. I used an export of the Pokec social network denoted as PKC30k (30,002 nodes, 245,790 edges), which is the most popular online social networking site in Slovakia. I furthermore used an export of the Facebook social networking site denoted as FB30k (30,002 nodes, 593,476 edges). Finally, I also included a fragment of the DBLP co-author network denoted as DBLP80k (80,002 nodes, 602,096 edges). DBLP is bibliography website on computer science, and relation between entities are citations between different works. The degree distribution of all the included networks are plotted in Fig. 4.4b, and the results on comparing the difference of LTA$_A$ and LTA$_{deg}$ after running the same simulations are displayed in Fig. 4.5b.

These measurements verify the intuition on why LTA$_A$ provides better performance in LJ66k dataset, as results are similarly better also in the DBLP80k, FB30k, PKC30k networks. To put the difference between LTA$_A$ and LTA$_{deg}$ into another perspective,

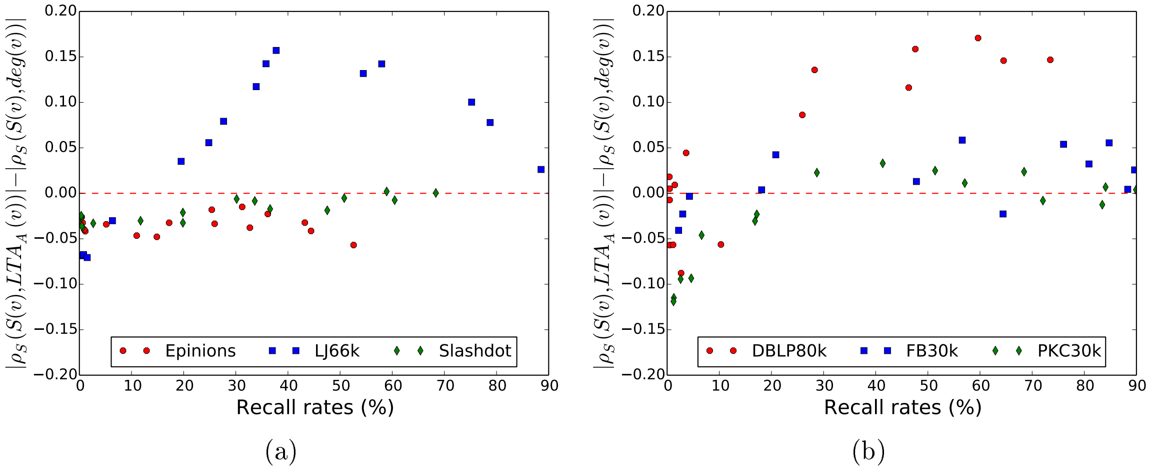


Figure 4.5: Difference between the correlation values of LTA$_{deg}$ and LTA$_A$. From (a) it is visible that in LJ66k LTA$_A$ has better results than in the other networks. Comparing correlation in further networks having similar degree distribution to LJ66k is shown on (b). In these cases LTA$_A$ proves to be again a better measure for anonymity.

we could interpret the LTA measures as node fingerprints. Node degree is a first level node fingerprint with a limited information on the node neighborhood, while LTA$_A$ is a second level node fingerprint that incorporates information from the neighborhood of the node at a distance of $d = 2$. Looking from this perspective, we can expect LTA$_A$ to perform better in cases where the degree distribution is more balanced, and differences in degree values are less prominent.

Evaluation of Measures for the Grasshopper algorithm

With the Grasshopper attack, I measured the ranking property of the LTA$_A$ and LTA$_{deg}$ variants on the same datasets used to provide the results in Table 3.1. The results are shown in Fig. 4.6, where each dataset is distinguished by color and variants are plotted with different marker (absolute value of correlation results are displayed).

Recall clearly has a significant effect on the correlation; however, the figure shows that both evaluated variants had acceptable correlation rates, and LTA$_{deg}$ had better results in general except for a few cases. Here, the network structure did not bias the results as in the case of Nar09. In summary, these measurements revealed that the concept of LTA can be applied for other algorithms as well.

4.1.3 Conclusion

Regarding the proposed measures presented in this section and related experiments, we can conclude that both LTA$_A$ and LTA$_{deg}$ can be used as apriori relative anonymity measures without knowing the background knowledge of the attacker. In addition,

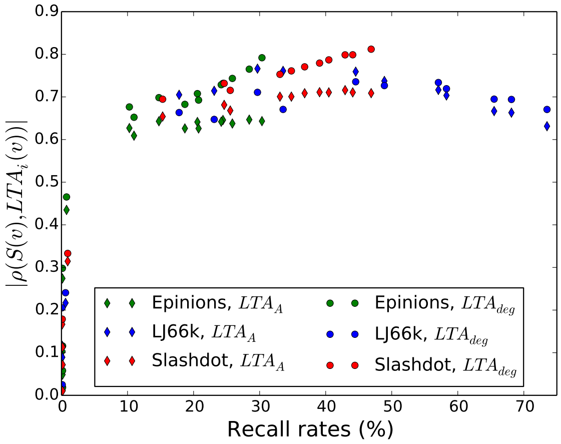


Figure 4.6: Measuring correlation between output of LTA variants and re-identifications rates of Grasshopper on nodes. Recall clearly has a significant effect on this, however, the figure shows that both evaluated variants had acceptable correlation rates.

these measures are designed to be flexible and can be changed in the future, e.g., when being used with novel attacks. Comparative experiments with these measures showed that in distinguished cases each is better; however, in general, it seems to be more feasible to work with LTA_A, as in life-like networks with a biased degree distribution is has significantly better correlation rates with re-identification than node degree.

4.2 Measuring the Importance of Seeding

Related to the effect of seeding on propagation, Narayanan and Shmatikov highlight in [2] that seeding has a phase transition property regarding the number of seeds [59]: at some point while increasing the number of seeds, there is only a little difference when the output of propagation rises significantly, reaching the maximum (example provided in Section 3.4 in Fig. 3.2b). They also note (without details) that transition boundaries depend from networks structure and seeding method. Seeding stability is also mentioned in their paper as the probability of large-scale propagation with respect to the number of seeds. However, beside their findings related works focusing on seeding is only of a few papers, and in most papers with simulations the chosen seeding method is not justified.

Yartseva and Grossglauser provide further analysis of seeding [70], and they propose two simpler, but similar algorithms to Nar09, that allow formal analysis. In their work, the existence of phase transition is formally proven w.r.t. the seed set size for random graphs generated by the Erdős-Rényi model $G(n, p)$. Phase transition is also verified by simulations both for synthetic and real-life social networks. Their work discusses the essential seed set size for propagation as a function of the network parameters and the propagation settings, but neglects how seeds were obtained, i.e., the seed selection method.

While seed size and phase transition are studied aspects of the attacker model in the literature, there are still several questions left open. For instance, how strong is the difference between different seeding methods, e.g., w.r.t. minimum seed size and seeding time? Is there a globally best seeding method? We analyzed analyzed these and other related questions in our works in [C1, C3], and these results are presented in this section.

4.2.1 Review of Seeding Methods for Evaluation

The seeding method reflects the strength of the attacker, who is often limited by the quality of the background knowledge he has. However, a well-informed attacker may have the opportunity to choose between different seeding methods.

Before to our evaluation in [C1], there were several seeding methods appearing in the literature. The original paper used high-degree nodes for seeding that formed 4-cliques [2], and in addition to this, in their main experiment they used seed nodes with at least a degree of 80. Several other seeding methods appeared in the literature, as matching top nodes [1, 14], (presumably) sampling random nodes in [15], and seed selected randomly from top 25% high degree nodes [C2]. In their work of Srivatsa and Hicks they used betweenness centrality for seed selection in G_{src} and proposed a similar probabilistic variant of a distance measure to find corresponding nodes in G_{tar} [10].

In the current experiments I generalized clique based methods, where seed nodes were requested to form *k-cliques* ($k \in \{4, 5, 6\}$). Furthermore, I had cases where node degree was not considered (later referred to as e.g., `4cliques`), while in other cases seeds were sampled from the top 20% by degree (e.g., `4cliques.2`). In order to see the magnitude of the effect of the clique structure, I also compared these results against *k-neighborhood* seeding (with corresponding parameters), where nodes are collected with breadth-first search starting from a random node (e.g., `4bfs`, `6bfs.2`).

In order to see how degree itself influence overall results, I included using *k-top* degree nodes (`top`), and sampling from *random high degree nodes* in the top $10\%, 25\%, 50\%$ subsets (e.g., `random.25`), and from all nodes (`random`), for the sake of completeness.

I also analyzed more complex measures than node degree, namely *betweenness* (e.g., `betwc.2`, seeds that had the highest betweenness in the set of the top 20% by degree) and *closeness centrality* (e.g., `closec.2`). These measures can be calculated together as being based on shortest paths: betweenness reflects centrality respecting the number of shortest paths the node is on, while closeness gives the average distance from all other nodes in the network. Betweenness was used only for small networks in [10], thus its utility in larger networks was uncertain until our paper in [C1]. In addition, calculating betweenness and closeness is very costly for large networks, hence we also analyzed if the number of nodes involved in the calculation process can be decreased (by using degree as a heuristic).

I included two additional exotic seeding measures. As it was discussed recently, *Local Topological Anonymity* values are calculated according to the structural uniqueness of nodes in their 2-neighborhoods: the lower the value the more unique the node is. Thus, the intuition was that nodes with low LTA_A values are likely to be good seeds (marked as `lta`). I also tested seeding with nodes having the highest *Local Clustering Coefficient* (LCC) values in the network (`lcc`). Here, I had the intuition that probably not the nodes with the most dense neighborhood are providing the better seeds, hence I measured high LCC (`lcch`), where highest LCC nodes were selected after skipping the top 20% of LCC.

4.2.2 Evaluation of Seeding Methods

During the evaluation I calculated measures on G_{src}, and to keep the focus on comparison of measures, I used the ground truth to map selected seed nodes to their pairs in the G_{tar}; instead of constructing new methods that could be used in realistic situations. However, these methods can be implemented to work without background knowledge, there are several examples of such implementations in the literature [2,10, 14].

Within the experiments, I was looking to find the minimum number of seeds granting large-scale propagation in all measurements (which can be called as stable seeding), and measured runtime of the seed selection phase (or in other words, time resource requirements). It could be interesting to compare recall rates for different methods, but I could only find minor differences in those. I used seed size stepping granularity as 5 in simulations executed on LJ10k and 60 in larger networks (or lower for competitive techniques in order to get more detailed measurements).

These experiments were run on a 2.0GHz Intel Core i7 processor with 8GB RAM, and my framework was implemented in Java (having almost 8,000 lines of code). However, it must be noted that I intended to show differences between measures and highlight trends, and not to provide razor-sharp results; thus I caution drawing conclusions from subtle differences in results (e.g., minimum seed set sizes of `betwc.1` and `betwc.25` in Slashdot in Fig. 4.8b).

It must be noted regarding runtimes that some measures require significant preliminary calculations to seeding: betweenness and closeness centrality (these can be calculated in parallel), LTA, and LCC. I did not include these preparations into seed timings, as although they may run longer, yet these are still computationally feasible (e.g., within a few hours of computation time), and need to be done once. Nevertheless,

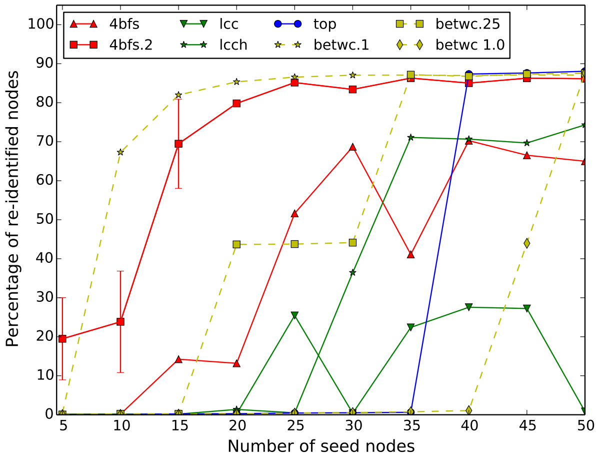


Figure 4.7: Differing characteristics of seeding strategies in LJ10k.

an attacker may consider this when choosing the seeding method.

Large-Scale Propagation

Initial simulations were performed on LJ10k. Results revealed that node degree takes an important position as a secondary measure of seed selection. For all k-clique and k-neighborhood based methods it could be observed that when using high degree seeds, less nodes are needed for large-scale propagation, and more importantly, Nar09 was able to access the network more widely. For instance, compare `4bfs` and `4bfs.2` in Fig. 4.7 – there is a clear limit for propagation when using `4bfs` seeding. Thus I used only high-degree variants of the k-clique and k-neighborhood seeding methods in the analysis related to larger networks. Fortunately, this has additional benefits, as it speeds up seeding.

Degree dependent node selection for other measures also lead to differences in results, although it did not limit the maximum level of propagation. The examples shown for betweenness centrality in Fig. 4.7 illustrate how degree defines the number of seed nodes that are required for successful propagation.

Other factors can influence results also. While `lcc` could not reach an acceptable level of re-identification in these measurements (resulting recall rates at most around 20%), the `lcch` variant produced better rates, though it was also incapable of reaching recall rate significantly higher than 70%, similarly to `4bfs` (this is clearly visible in Fig. 4.7).

Additionally, measurements in Fig. 4.7 confirm that phase transition property of the propagation phase depends on the seeding measure (as also mentioned in [2]): phase transition start- and endpoints, steepness differ for various methods. For example, while phase transition both for `4bfs` and `4bfs.2` start early, and have a mild increase, for the `top` method it can be rather characterized as a sharp jump.

Seed Stability

Example for `4bfs.2` on the LJ10k network provides insight on seeding stability in Fig. 4.7. While it allows propagation reaching high-end of recall for $|\mu_0| \in [20, \ldots, 50]$, it can even achieve an average recall of 20% for 5 seed nodes. By including the variance besides (normalized by multiplying it with 10^{-2}), a notable variance can be noticed initially, taking values between 1000-1300. As seeding gets stable, it apparently disappears as it takes values between 0.1-23.3 for $|\mu_0| > 15$ (other experiments showed similar behavior, but these are not displayed for keeping the figure clear). This happens for a simple reason: in case of such a small amount of seeds the current instance of seed nodes determines significantly the overall outcome of Nar09, e.g., in these experiments propagation achieved recall rates of 0.26% or 78.1% for different seed sets.

As the error rate is low by design, an attacker can settle with a low number of seeds that leads to large-scale propagation. This even works in larger networks: with only a single 5-clique seeding (`5cliques.2`) where I could achieve recall rate as high as 84.33% having the error rate at 5.62%.

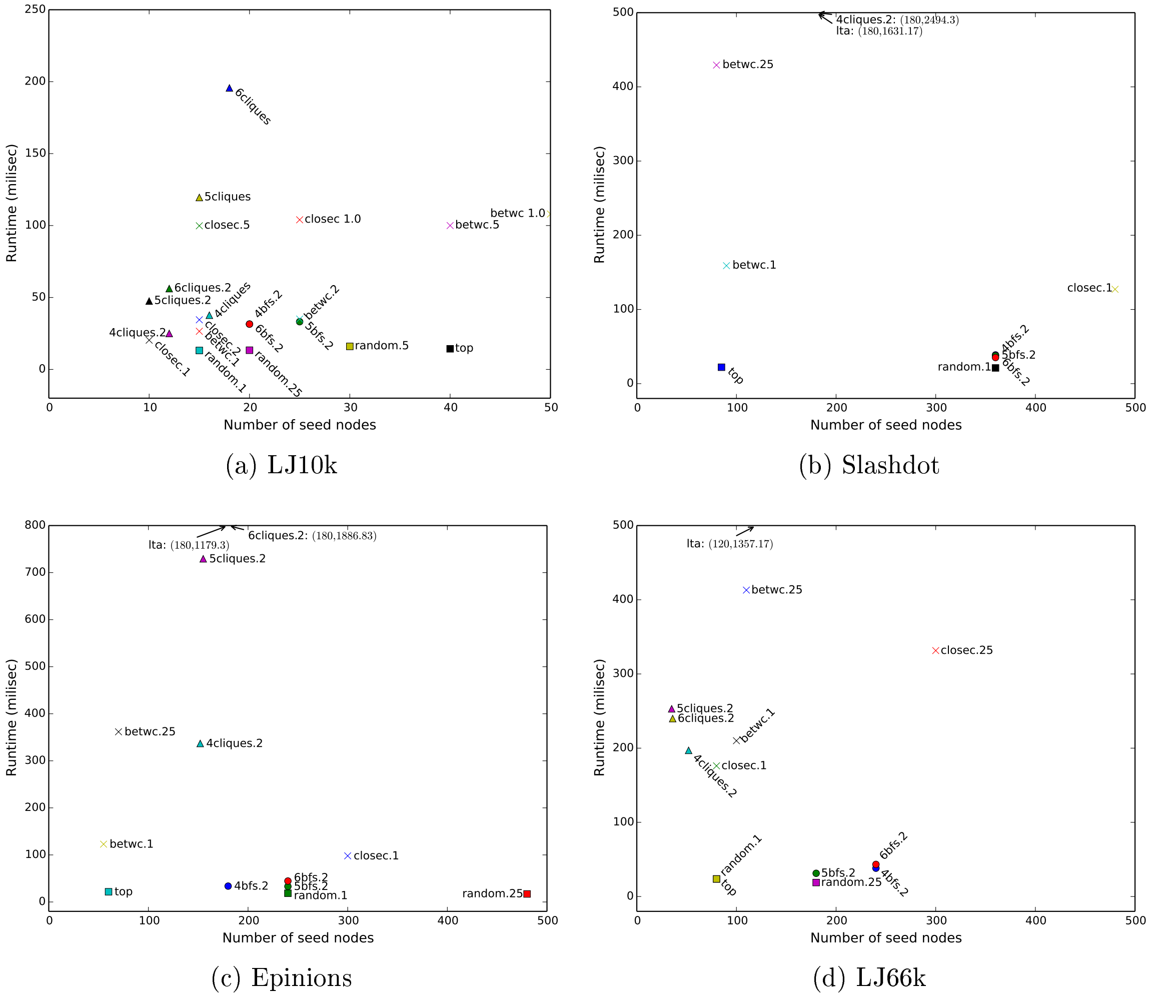
Degree as a Heuristic

The summary of the measurements for LJ10k and the main three networks is shown in Fig. 4.8, including methods that resulted in large-scale propagation, and where runtimes and the number of required seeds were sufficiently low. With accordance of the results in LJ10k, where these measures with higher degree nodes resulted better recall rates, I only calculated betweenness and closeness centrality on high-degree nodes to reduce runtimes (top $10\%, 25\%$). For all three networks results showed that the higher degree nodes were used, the lower the seeding time was.

k-cliques and k-neighborhoods

The network structure determines which seeding methods could be used or not. Using cliques were not feasible in the Slashdot network: although it was possible to find enough seeds with `4cliques.2`, this was a less prominent result. In addition, for `5cliques.2` and `6cliques.2` the seeding algorithm timed out (2 mins) before finding enough disjoint cliques. These methods were more competitive in the Epinions network, best results were measured in the LJ66k dataset, as these were capable of stable seeding with the least number of seeds.

For reaching unstable large-scale propagation in the networks originated from LiveJournal, cliques provided also very competitive results. This is likely due to the fact that the connectivity of these networks is more balanced than in the others, as they have proportionally less low degree nodes than in the others (see Fig. 4.4b in Section



(a) LJ10k

(b) Slashdot

(c) Epinions

(d) LJ66k

Figure 4.8: Performance of propagation phase vary as different seeding methods are used. While some methods performed equally well in all cases (e.g., `betwc.1`), some methods produced different results according to the size of the network (e.g., `top`), to structural differences (e.g., `4cliques.2`), or according to the relationship between seed nodes (e.g., `4cliques.2` compared to `4bfs.2`).

4.1.2). For example, a single clique (of any size) was enough in LJ10k to have large-scale propagation, and Nar09 could achieve the highest recall level by simply using two cliques. In LJ66k, a single `4cliques.2` was enough to reach recall of 33.32% with an error rate of 4.34%. Comparing these results to `4bfs.2`, `5bfs.2` and `6bfs.2` shows that structure between seed nodes can make a perceptible difference in the performance of propagation. In addition, the latter techniques were not sensitive to network structure: these had low runtimes in all large test networks, but also had an average score regarding seed sizes.

Most Effective Methods

Comparing results plotted in Fig. 4.8, clearly `top` and `betwc.1` seeding methods led to best results, that were additionally independent of network structure (in larger networks). The discovery of betwc.1 in this context is important, e.g., as a protective method may aim preventing de-anonymization by targeting top nodes, either by removing or modifying them. Thus betwc.1 allows the attacker choosing seeds from a larger candidate set. The random.1 method is slightly less effective, but it could also be used alternatively. Additionally, closec.1 provided remarkably good results in the densest test network.

Exotic Seeding Measures

None of the exotic seeding methods could be emphasized for providing good results. Regarding the minimum number of nodes required for stable seeding, the lta measure produced fair results in large networks, but due to the large number of nodes it worked with it had high runtimes. The lcc and lcch seeding methods had even worse results; both only led to noticeable propagation in LJ66k, and had long runtimes. However, we could not include these results as their highest recall rate was less then the maximum (as in LJ10k).

4.2.3 Conclusion

The evaluation of multiple seed selection methods on the Nar09 algorithm showed that the chosen method can significantly influence and limit the possible outcome of the propagation. Experiments showed that both the global role of the seed nodes (measured with betweenness, closeness, degree) and the local structure between them (clique structure vs. k-neighborhood) can solely and jointly determine the success of propagation with the given seeding. In addition, results indicate that the seeding procedure should be chosen regarding network size and structure, as not all methods worked equally well for all datasets: phase transition boundaries also depend on the chosen method and datasets.

However, these results have greater importance and not limited to this dissertation. I believe my findings are essential for works aiming to compare novel attack techniques to others and for papers including simulation evaluations of defense methods. For the prior, it is needed to synchronize attacker models, including the seeding method in order to settle down on the same ground for comparing results. In the latter case, seeding methods represents another aspect of the attacker model that can be tuned for alternative (and stronger) attacks. For example, an attacker can react by choosing another seeding procedure in order to decrease the performance of the users of a given privacy-enhancing technique.

Therefore I used multiple seeding mechanisms in the evaluation of identity separation, as a part of the attacker model. By default, I applied random seed selection with high degree nodes, where nodes are selected from the top 25% by degree (denoted as `random.25`). For the simulation of stronger attackers I used seeding methods that proved to be the most effective: sampling from higher degree nodes (`random.1`), accurately selecting top degree nodes (`top`), and nodes with the highest betweenness values also having degree in the top 10% (`betwc.1`).

Chapter 5

Analysis of Identity Separation

> 'Privacy is an inherent human right, and a requirement for maintaining the human condition with dignity and respect.' (Bruce Schneier)

In this chapter I analyze the effect of the identity separation technique on structural re-identification, both regarding global node identification and propagation phases of Nar09. In the first case, I show suitable strategies for two seed identification techniques that lead to high failure probabilities for the attacker even when users do not cooperate. Similarly as provided, models proposed in Section 3.5 can be used for the evaluation of other seeding and node re-identification methods, also. For the propagation phase I both consider the non-cooperative and multiple cooperative settings for adopting identity separation, and analyze with simulation experiments their effect regarding user and network privacy.

The outline of this chapter is as follows. I present the analysis of seeding with respect to identity separation in Section 5.1. Then, in Section 5.2 I analyze the sensitivity of the propagation phase to different features of identity separation, then continue my analysis when the technique is adopted in a non-cooperative fashion in Section 5.3. Then I analyze when neighboring users cooperate in Section 5.4, and the efficiency of globally organized cooperation in 5.5. Last, but not least, I highlight the importance of the participation of top degree nodes in Section 5.6 for most of the discussed cases, and the chapter is finally concluded in Section 5.7.

5.1 Characterizing Failure Probability for Seeding

In this section I show examples how seed methods can be analyzed by using the probabilistic identity separation models in Section 3.5.5. Thus, I provide formal analysis of the two seeding methods: clique based seed selection introduced in [2] and another one based on top nodes that is used in multiple works (e.g., [1, 14]). These results can serve as a guideline extending the underlying principles to other cases of global node identification, too. Here I consider the seed candidate nodes to be using identity separation independently – as with cooperation these nodes could achieve their goal more easily. For example, cliques could be more easily destroyed with cooperation.

Here, I use the basic and the realistic models, which are the closest to real user behavior: users are expected to have roughly the same number of contacts before and after the identity separation (not including new contacts). However, the analysis could be extended to the worst and best models also, and results for the best model could be deducted from the results of the realistic model, as higher failure can be expected.

5.1.1 Active Seeding and Identity Separation

Backstrom et al. describe two attacks, a semi-passive and an active attack, in which both the attackers are able to modify the network prior to the sanitization [13]. In both attacks the attackers' goal is to insert a specific structure (a subgraph) into the social graph that can be revealed later only by the attackers but no one else – this is what they call structural steganography. This subgraph is connected to the social graph by creating new edges to a small number of targeted users. A similar approach including the injection of a subgraph constructed by the attacker is proposed for seeding the seed-and-grow attack [15].

These works reveal one the disadvantages of active seeding: it only allows revealing the identity of a small number of users, who are willing to play along with the attacker during the injection phase. In addition, for some networks active and semi-passive seeding cannot be executed at least for one of the following reasons:

- The modification of the network structure may be expensive (e.g., phone calls).
- The modification may not be executable (e.g., network created from observed e-mails).
- To insert the structure too many modifications would be required (e.g., a valid e-mail address must be provided for the registration).
- The attacker is not always able to influence connections (e.g., connections require two-way confirmation).

All these problems inspired the research of passive seeding methods, such as the clique based one proposed in [2]. As passive methods can be considered more lifelike, and could be used in a wider range of scenarios, the following analysis focuses on those.

However, from the viewpoint of identity separation, it is easier to protect the privacy of a user against active seeding methods. While active attacks cannot be prevented, one can use identity separation to separate sensitive information from suspicious users. Even a complete neighborhood can be separated to prevent possible identification of a sensitive partial identity.

5.1.2 Generic Formula for Failure Probability

It must be noted that different actors have different views on the measuring success. The adversary is interested in discovering the correct mapping for several nodes. As such, he is likely to be interested in the probability of success in identifying a set of $|\mu_0|$ seed nodes. The point of view of a user is, on the other hand, that he himself should not be vulnerable to the attack, and other users are more or less irrelevant to his. Also, a user is rather interested in the failure probability of the attacker. This is why I have focused on calculating failure probability of single users; however, based on my analysis, attacker success rates could be also estimated.

The probability of failure of seeding for a node v_n^{tar} (with $N = \deg(v_n^{tar})$), based on the previously discussed notation and assertions, can be described by using the law of total probability as

$$P(\text{"failure"}) = P(Y=0) + \sum_{y=1}^{N} P(\text{"failure"}|Y=y) \cdot P(Y=y). \tag{5.1}$$

The first member is the probability of the case where the user has 0 identities in the exported graph, i.e., all of his edges are anonymized and thus removed. The other part is the sum that incorporates General Assertion 2, namely that the user can create at most as many identities as many contacts he has. The results for the different submodels of user behavior mainly deviate in the definition of the conditional probability $P(\text{"failure"}|Y = y)$. Note that the formula for the sum may slightly also differ in some cases, e.g., in that of the basic model, where it does not include probabilities for $Y = 1$, as in that case the attack should not fail, as the original identity is impeccably preserved in all cases.

In the general case, the conditional failure probability $P(\text{"failure"}|Y = y)$ can be unfolded as

$$\begin{aligned} &P(\text{"failure"}|Y=y) = \\ &\qquad = \sum_{\forall X} P(\text{"failure"}|X_1=x_1,\ldots,X_y=x_y) \cdot P(X_1=x_1,\ldots,X_y=x_y), \end{aligned} \tag{5.2}$$

where $X = (x_1, \ldots, x_y)$ represents a given layout for identity separation. Failure probability can be furthermore characterized depending on the seeding method and the identity separation submodel.

5.1.3 Clique-Based Seeding Method

The first passive seeding method that required no background knowledge, was introduced in [2]. First, the attacker picks a 4-clique from G_{src}, then computes the degree of each vertex and the number of common neighbors for each pair of nodes. Next, he looks for similar 4-cliques with similar values (within a factor of $1 \pm \epsilon$) in the target graph. The error factor is considered for mapping each vertex (in the case of degrees) and each pair of vertices (in the case of common neighbor counts). This can be further generalized for k-cliques.

Structural modifications within the cliques are thus disallowed; identification fails if any of the edges are removed from the clique. While the original algorithm compares common neighbor counts as well, the following analysis shows that even these two criteria can be violated effectively with identity separation (i.e., the following is a lower bound for the failure probability).

Although here only the failure probability of a single node is defined, but for a clique it can be calculated simply by giving the probability of the union of failure events for members. Therefore the calculation does not take actions of other users in the clique into account, meaning that it is assumed that they neither perform identity separation, nor anonymize any of their edges. If we take these effects into account, the failure probability would be higher in most cases, and at least equal, since other users could also destroy the clique or change the degrees of the vertices thereof, making identification less probable.

Preliminary Naïve Analysis on 4-cliques

The goal of this preliminary analysis is to determine the possible effect of identity separation on cliques in the network. Cliques can be destroyed if:

- One of the members separates himself totally from the clique. This is equivalent to the removal of the representing node.
- One of the members removes at least one internal edge from the clique.
- One of the members separates at least an edge from the clique (but this edge will not be deleted).

I executed simulation experiments to determine how effectively identity separation removes 4-cliques from the network. For the experiments, I sampled $10,000$ nodes from the Slashdot network containing $1,816,110$ 4-cliques, and $1,000$ nodes were sampled containing $2,102,842$ 4-cliques from the Epinions network. For comparison, a full graph of 100 nodes (with $3,921,225$ 4-cliques) was also included.

The basic model with uniform edge sorting probabilities and $Y = 2$ were used for the simulation. We can define a theoretical limit to show the expected number of cliques affected by identity separation. Adding more privacy-enhancing functionality to the simulation (compared to the basic model), such as edge and node removal, the number of cliques would be furthermore decreased, closer to the theoretical limits.

Given a k-clique $C_k = \{v_1, \cdots, v_k\} \in G_{tar}$, the probability if there are any identity separations is

$$\begin{aligned} P(\text{'id.sep. in } C_k\text{'}) &= P(\exists v_i \in C_k : y_i > 1) = \\ &= 1 - P(\forall v_i \in C_k : y_i = 1) = 1 - P^k(Y = 1), \end{aligned} \tag{5.3}$$

where y_i denotes the number of identities created by v_i.

Therefore, the expected number of cliques remaining intact can be calculated as the expected value of the binomial distribution $N_{intact} \sim B(N_{cliques}, P^k(Y = 1))$, where $N_{cliques}$ denotes the number of 4-cliques in the original graph. The expected value of N_{intact} is

$$E[N_{intact}] = N_{cliques} \cdot (1 - P(\text{'id.sep. in } C_k\text{'})) = N_{cliques} \cdot P^k(Y = 1) \tag{5.4}$$

The relative values of $E[N_{intact}]$ with $k = 4$ are denoted in Fig. 5.1 as the expected number of cliques remaining intact by identity separation (denoted as Theoretical limit). It is possible that the clique remains a clique, but the probability of recovery depends on further errors regarding the compared degree and common neighbor count values.

According to the experiments, as the number of users who use identity separation increases, the number of 4-cliques decreases fast and almost in the same pace for all test datasets (see Fig. 5.1). For instance, in both test networks for $P(Y = 2) = 0.2$ the number of remaining cliques was almost halved: the percentage of intact 4-cliques was 52.26% for the Slashdot sample, 51.27% for the Epinions sample, and 55.22% for the full graph. It is also visible in Fig. 5.1 that graphs having more 4-cliques degrade slightly faster, for a simple reason: usually several 4-cliques overlap in a single node, and therefore splitting it causes the deletion of multiple 4-cliques. We can conclude the naïve analysis is in that identity separation erodes network structure effectively, thus it needs to be furthermore analyzed.

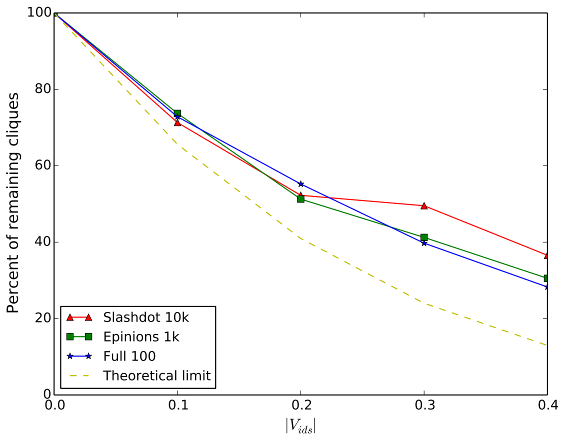


Figure 5.1: Simulation results (including the theoretical limit) show the degradation in clique numbers in case of allowing identity separation.

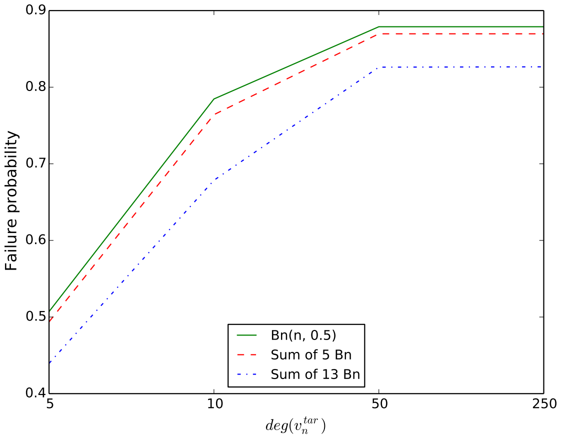


Figure 5.2: Failure probabilities in the realistic model, with clique seeding, for different distributions with $Y = 2$, for different sizes of $\deg(v_n^{tar})$.

Modeling Clique-based Seeding

As identity separation can have different effects on the resulting error depending whether internal or external edges were separated or removed, this should be included in the model. Node $v_n^{tar} \in V_{ids}$ is a user who is part of a k-clique C_k, and has $N = \deg(v_n^{tar})$ neighbors in G_{tar}, and therefore node v_n^{tar} has $k-1$ internal and $n-k+1$ external edges, as seen from the viewpoint of the clique. For the inner edges, the distribution of the edge sorting is described as $P(X_1 = x_1', \ldots, X_y = x_y')$, with no predefined distribution included (distributions are defined with the chosen model). For the outer edges, the distribution is described similarly as $P(X_1'' = x_1'', \ldots, X_y'' = x_y'')$. X_i' and X_i'' are random variables describing the number of edges between the i^{th} identity and the members of the original clique, and those between the i^{th} identity and the neighbors of the original node, respectively.

The clique based seeding involves an error parameter ϵ for the seed identification, and an error measure based on it: the compared node degree values need to match within an error factor of $1 \pm \epsilon$. Based on this, we can now define an error measure function that can be used in the calculation of the failure probability for node v_n^{tar} in the clique C_k:

$$e(x_i', x_i'') = \begin{cases} 1 & \text{if } (\frac{x_i'+x_i''}{N} < 1 - \epsilon \land x_i' = k-1) \lor x_i' < k-1 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

where x_i'' denotes the number of outer, and x_i' the number of inner edges for a given identity.

The original node degree value N, the clique size k, and the error parameter ϵ are assumed to be known constants. The clique size and the error parameter are new attacker-dependent parameters, who can manipulate these in order to achieve better results, and also several attacks with different values can be executed for these parameters, without any limitations.

Formula of Failure Probability for Clique Based Seeding

Here I provide the calculation of the lower estimate of failure probability for a single clique. Fortunately, there are user settings of p_i, where this is so high that even if the user is member of multiple cliques, failure is yet expected to happen.

Continuing the elaboration of (5.2), probability $P(\text{"failure"}|X_1 = x_1, \ldots, X_y = x_y)$ can be calculated differently for two cases. First, we need to distinguish between internal edges as **X'** $= (x'_1, \ldots, x'_y)$ and external edges as **X"** $= (x''_1, \ldots, x''_y)$, where counts originally are respectively $|\mathbf{X'}| = k - 1$ and $|\mathbf{X''}| = n - k + 1$. In the first case $\forall x'_i < k - 1$, the clique is always destroyed, since all edges are sorted in groups having less than $k - 1$ edges, then $P_{clique}(\text{"failure"}|X'_1 = x'_1, \ldots, X'_y = x'_y) = 1$ always. In the other case, where $\exists x'_i = k - 1$, $P_{clique}(\text{"failure"}|X'_1 = x'_1, \ldots, X'_y = x'_y)$ is calculated as

$$\begin{aligned} P_{clique}(\text{"failure"}|X'_1 = x'_1, \ldots, X'_y = x'_y) = \\ = P\Big(\bigcup_{\forall X''} (X''_1 = x''_1, \ldots, X''_y = x''_y \mid e(x'_i, x''_i) = 1)\Big). \end{aligned} \tag{5.6}$$

By knowing that these events are mutually exclusive, this equals to

$$\begin{aligned} P_{clique}(\text{"failure"}|X'_1 = x'_1, \ldots, X'_y = x'_y) = \\ = \sum_{\forall X''} \Big(P(X''_1 = x''_1, \ldots, X''_y = x''_y) \cdot e(x'_i, x''_i)\Big). \end{aligned} \tag{5.7}$$

Therefore, in general for cliques, the failure probability for node v_n^{tar} with y identities can be described as

$$\begin{aligned} P_{clique}(\text{"failure"}|Y = y) = & \sum_{\forall X': \nexists x'_i = k-1} P(X'_1 = x'_1, \ldots, X'_y = x'_y) \\ & + \sum_{\forall X': \exists x'_i = k-1} P(X'_1 = x'_1, \ldots, X'_y = x'_y) \\ & \cdot \Bigg(\sum_{\forall X''} \Big(P(X''_1 = x''_1, \ldots, X''_y = x''_y) \cdot e(x'_i, x''_i)\Big)\Bigg) \end{aligned} \tag{5.8}$$

Analysis of the Basic Model

In this case, the basic model can be evaluated by using Multinomial distribution as proposed in Section 3.5.5. The formula for failure probability can then be derived similarly to (5.8) as:

$$
\begin{aligned}
&P^{B}_{clique}(\text{"failure"}|Y=y) = \\
&\qquad \sum_{\forall X':\nexists x'_i=k-1} P(X'_1=x'_1,\dots,X'_y=x'_y) \\
&\qquad + \sum_{\forall X':\exists x'_i=k-1} \Bigg(P(X'_1=x'_1,\dots,X'_y=x'_y) \\
&\qquad \cdot \Big(\sum_{x''_1=0}^{n-k+1} \cdots \sum_{x''_{y-1}=0}^{n-k+1-\sum_{j=1}^{j=y-2} x''_j} P\big(X''_1=x''_1,\dots,X''_y=n-k+1-\sum_{j=1}^{j=y-1} x''_j\big) \cdot e(k-1,x''_i)\Big)\Bigg)
\end{aligned}
\tag{5.9}
$$

where $\sum x'_i = k-1$ is in each sum. Using the formula and properties of the multinomial distribution this can be simplified as:

$$
\begin{aligned}
&P^{B}_{clique}(\text{"failure"}|Y=y) = \\
&\qquad 1-\sum_{\forall i\in[0,\dots,y]} p_i^{k-1} + \sum_{\forall i\in[0,\dots,y]} \Bigg(p_i^{k-1}\cdot \\
&\qquad \cdot \Big(\sum_{x''_1=0}^{n-k+1} \cdots \sum_{x''_{y-1}=0}^{n-k+1-\sum_{j=1}^{j=y-2} x''_j} P(X''_1=x''_1,\dots,X''_y=n-k+1-\sum_{j=1}^{j=y-1} x''_j) \cdot e(k-1,x''_i)\Big)\Bigg)
\end{aligned}
\tag{5.10}
$$

This can be further simplified as:

$$
\begin{aligned}
&P^{B}_{clique}(\text{"failure"}|Y=y) = \\
&\qquad 1+\sum_{\forall i\in[0,\dots,y]} p_i^{k-1}\cdot \Bigg(\sum_{x''_1+\dots+x''_y=n-k+1} \Big(\frac{(n-k+1)!}{x''_1!\cdot\ldots\cdot x''_y!} \cdot p_1^{x''_1}\cdot\ldots\cdot p_y^{x''_y} \cdot e(k-1,x''_i)\Big) - 1\Bigg)
\end{aligned}
\tag{5.11}
$$

Then, the overall failure probability can then be derived easily. We know that for $P^{B}_{clique}(\text{"failure"}|Y=1)$ the neighborhood of the node would remain the same, and therefore would not introduce any error in the seed identification. Using this, the

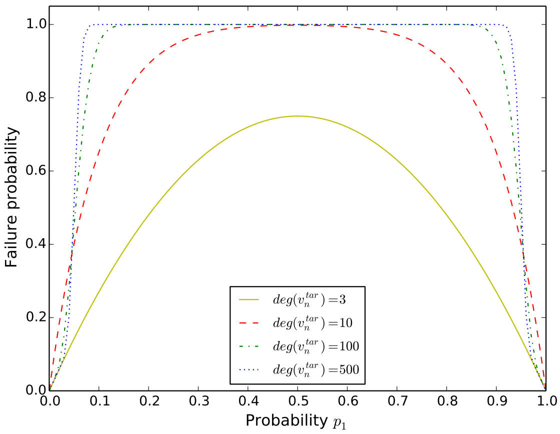


Figure 5.3: Basic model parameter analysis of $\deg(v_n^{tar})$: $P^B_{clique}(\text{"failure"}|Y = 2)$ as a function of p_1, with fixed $k = 4$ and $\epsilon = 0.05$ with different values for degree.

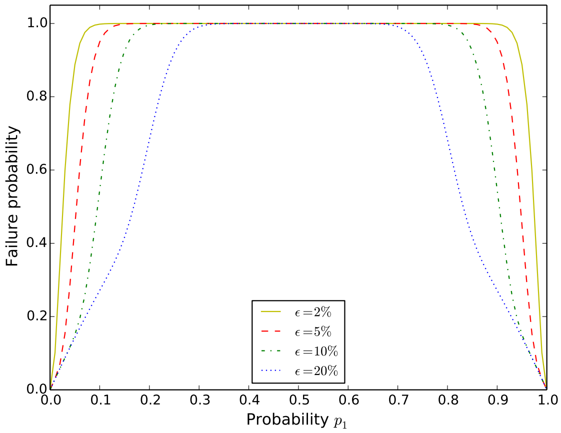


Figure 5.4: Basic model analysis of ϵ: $P^B_{clique}(\text{"failure"}|Y = 2)$ as a function of p_1, with fixed $k = 4$ and $\deg(v_n^{tar}) = 100$ with different values for ϵ.

probability failure formula now can be deduced from the general formula (5.1), with the exclusion of the case for $Y = 1$.

Fig. 5.3 describes how failure probability changes with different values for parameter N, while parameters $Y = 2$, $k = 4$ and $\epsilon = 0.05$ are fixed. The results have several interesting consequences. First of all, it can be seen that the failure probability is conveniently high even for small N-s (e.g., $N \geq 10$). Secondly, users are given a relatively wide range of options for making their identification fail. Even if they use identity separation for just two identities, and the probability of using the second identity is small, the failure probability still remains high (e.g., for $p_1 = 0.1$, $N = 100$: $P^B_{clique}(\text{"failure"}|Y = 2) = 0.949$).

Fig. 5.4 describes how the failure probability changes in the function of ϵ while parameters $Y = 2$, $N = 100$ and $k = 4$ are fixed. The curves do not deviate significantly for other N values either. The shape of the curve suggests that if a user adopts identity separation in a reasonable way, the adversary cannot influence the success of the seeding. According to the original paper [2], the value of ϵ should be around 0.05, and a practical limitation of $0 < \epsilon \leq 0.1$ applies. For these values, users should choose p_1 and p_2 such that $0.2 \leq p_1, p_2 \leq 0.8$ (with $p_1 + p_2 = 1$), because this marks failure probabilities that are likely to be beyond the control of the attacker. Finally, I have also analyzed the effect of clique sizes (parameter k), which turned out to have no notable bias on the failure probability, even for different neighborhood sizes (with $\epsilon = 0.05$).

In general, it cannot be stated that using a larger number of identities will eventually lead to higher failure probability. We can think of $Y = 3$ with $p_1 = 0.001, p_2 = 0.001, p_3 = 0.998$ as a counterexample, where it is likely that the original node will be almost preserved, and therefore can be re-identified. However, during numerical analysis, I have found that in this model for a fixed p_i, the failure probability for two identities is the lower bound for failure probabilities with a higher number of identities that include p_i:

$$P^{B}_{clique}(\text{"failure"}|Y > 2) \geq P(\text{"failure"}|Y = 2) \tag{5.12}$$

I measured this to be true for $Y = 3, Y = 4$ and $Y = 5$, and it is likely to be true for $Y \geq 6$; I leave the formal proof as future work. However, if this can be proven to be true in general, that would be important for two reasons. On the one hand, the current analysis provides a lower bound for failure probability. On the other hand, it could facilitate the estimation of the overall failure probability as well, as based on (5.1) and (5.12) we can state the following (in the basic model):

$$\begin{aligned} P^{B}_{clique}(\text{"failure"}) &= \sum_{y=2}^{N} P^{B}_{clique}(\text{"failure"}|Y = y) \cdot P(Y = y) \\ &\geq \sum_{y=2}^{N} P^{B}_{clique}(\text{"failure"}|Y = 2) \cdot P(Y = y) \\ &\geq P^{B}_{clique}(\text{"failure"}|Y = 2) \cdot (1 - P(Y = 0) - P(Y = 1)) \end{aligned} \tag{5.13}$$

To sum it up, we can conclude that if the users use identity separation rationally, considering the influencing power of different parameters as discussed, the attacker has a low probability of identifying the nodes within cliques. This means that users need to separate their contacts into larger, but not necessarily equally sized groups. Therefore, this user behavior model can be suggested for users as a practical way to use identity separation, since it offers powerful protection if applied widely throughout the network.

Analysis of the Realistic Model

In this section, I discuss the analysis of the realistic model, which allows the user to make multiple copies of his contacts, beside also letting his to delete some of them. Here I used the binomial distribution, a distribution reflecting that it is likely that the number of all contacts after the identity separation is similar to that before, i.e., a few deletions and copies are likely, but major deviations are not. These distributions are detailed in Section 3.5.5.

Accordingly to the given distributions and the generic formula for failure probability, I have done the parameter analysis numerically. Its characteristics are similar to that of the basic model, and the preliminary results are satisfactory for this model, too (see

Fig. 5.2). We can conclude that the results are satisfactory even for small N's in all distributions under examination. However, it can also be seen that the higher variance we have, the larger the failure probability is. For this model it should also apply, that the higher number of identities one uses, the higher the failure probability should be (with the restriction that the chances of duplication of edges is not too high).

5.1.4 k-top Seeding Method

In this section I provide another example how probabilistic identity separation models can be used to calculate failure probability of seeding. Let us now consider a seeding method that considers the top k nodes and matches them for seeding (e.g., [1, 14]).

We could state that the attack fails, if node v_n^{tar} falls out of the top k nodes after adopting identity separation. In addition, it should be far enough from the top k according to the degree-based ranking of the nodes so that even if the attacker tries another parameter $k' > k$ (e.g., $k = 50$ and $k' = 200$), he should not be able to find v_n^{tar}. However, such a global matching of k nodes should not be feasible for large numbers of k, otherwise this approach could be used for network alignment instead of propagation. Based on a threshold κ limiting the largest expected size of k ($P(k > \kappa) < \epsilon$), term "far enough" can be defined depending on the network size and degree distribution. For partial identities of $v_n^{tar} \in G_{tar}$, we can say that the attack fails if $\forall v_{n\backslash i} : \deg(v_{n\backslash i}) < d_{border}$ where $P(d > d_{border}) \simeq \frac{\kappa}{|V_{tar}|}$, where $P(d)$ denotes the degree distribution in G_{tar} and d_{border} denotes the degree that separates a given proportion of high degree nodes from the rest of the network.

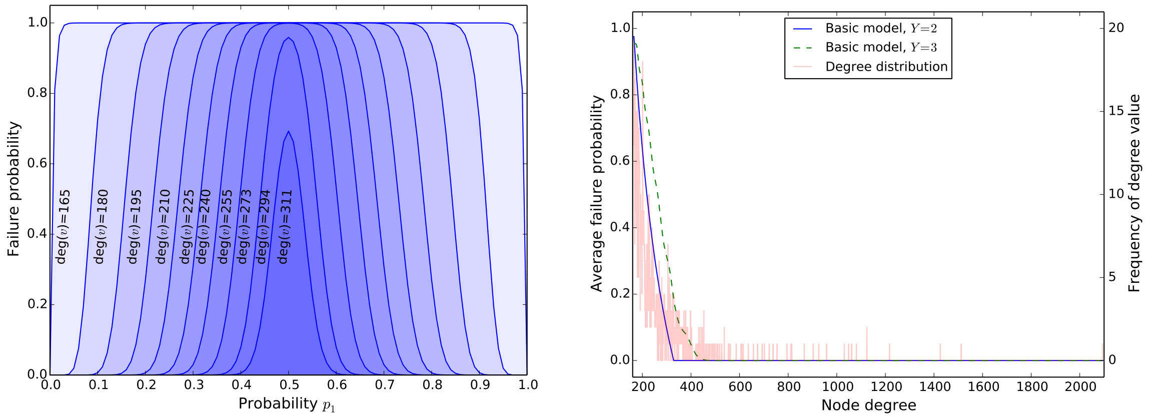
Based on this, we can introduce a function measuring error as

$$f(x_i) = \begin{cases} 1 & \text{if } x_i < d_{border} \text{ where } P(d > d_{border}) \simeq \frac{\kappa}{|V_{tar}|} \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

Then failure probability can be calculated as

$$P_{top}(\text{"failure"}|Y = y) = \sum_{\forall X} \Big(P(X_1 = x_1, \ldots, X_y = x_y) \cdot \prod_{i=0}^{y} f(x_i) \Big) \tag{5.15}$$

Regarding failure probabilities, the k-top seeding method has some significant differences compared to the clique based seeding. First, this seeding method only concerns a handful of nodes of the network. Second, it is harder for a node to make the seeding fail. This is for a simple reason: it is hard to decrease the degree of all partial identities under the desired threshold. For example, for a top node with $\deg(v) = 1000$ that uses



(a) Failure probability for k-top with the basic model, $Y = 2$.

(b) Average failure probability for k-top with the basic model, with different y-s.

Figure 5.5: Failure probability in the k-top seeding method, in the basic model. Even with $Y = 2$, for 80.4% of the top nodes ($165 \leq \deg(v) \leq 311$ with $\kappa = 1000$) have a significant failure probability for at least some settings (LJ66k network).

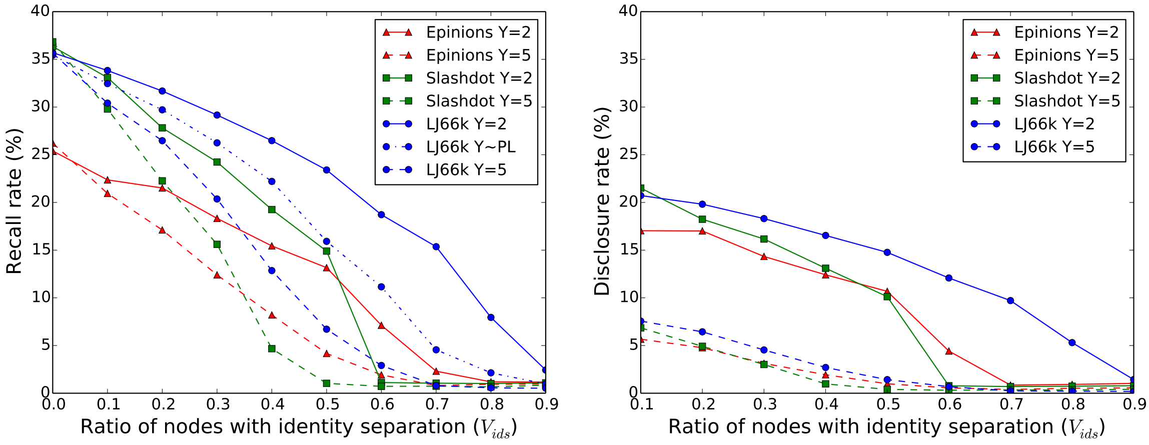
the basic model to have all node degrees under 300, should use at least $Y = 4$ to have the chance to succeed. However, these rates can be enhanced when deleting edges are also allowed (best model).

As an example, I have provided failure probabilities in the LJ66k network over the top nodes $\kappa = 1000$ in Fig. 5.5. Fig. 5.5a shows that even with $Y = 2$, for 80.4% of the top κ nodes ($165 \leq \deg(v) \leq 311$) have a significant failure probability for at least some settings, and the lower node degree is, the higher failure probability values can be measured. Fig. 5.5b shows how average failure probability values increase proportionally with y[1]. As a conclusion, these examples shows how the formula provided in (5.15) can be used to find the proper user strategy under the given circumstances. However, it must be noted that this is a lower estimate of failure probability: if a node still gets into the k-top that only means the possibility of re-identification, which is not necessarily to be successful.

5.2 Sensitivity Measurement of Propagation

In order to discover the strongest privacy-enhancing identity separation mechanisms, I investigated the efficiency of features in different models against the Nar09 algorithm. Initially measurements for recording recall rates with different perturbation as shown in Table 3.1 shown that the algorithm seems to be more sensitive to edge deletion than to node deletion, as the matrices for each dataset are not diagonal symmetric. Thus, in the following experiments, I analyzed these features of identity

[1] Granularity for the measurements was $p = 0.01$ for $Y = 2$, and $p = 0.1$ for $Y = 3$.



(a) Recall rates

(b) Disclosure rates

Figure 5.6: Experimental results using the basic identity separation model.

separation to see their effect separately.

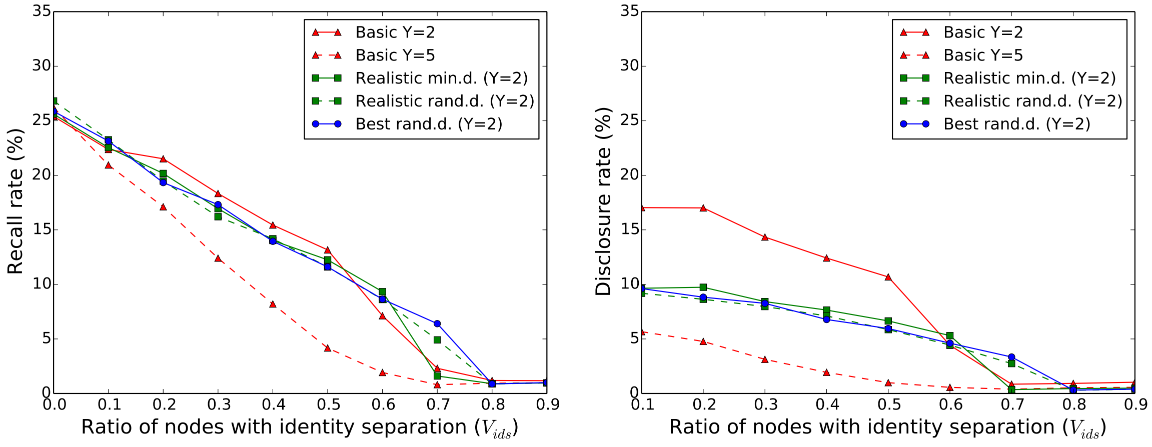
5.2.1 Characterizing Sensitivity to the Number of Identities

First, I tested the Nar09 algorithm against the *basic model with uniform edge sorting probability* on all networks having a ratio of users applying identity separation of $|V_{ids}| \in \{0.0, \dots, 0.9\}$. For the selected users a fixed number of new identities were created ($Y \in \{2, 5\}$). Results are summarized in Fig. 5.6, and while results for $Y \in \{3, 4\}$ not displayed, these can be easily interpolated.

Against my expectations, the basic model with $Y = 2$ and uniform edge sorting probability turned out not to be effective in stopping the attack. For the Epinions and Slashdot networks the recall rate mildly decreased until the ratio of privacy-protecting users reached circa $|V_{ids}| = 0.5$. For the LiveJournal graph the recall rate shows relevant fault tolerance of the attack (probably because of network structure, see Fig. 4.4b), e.g., 15.36% are still correctly identified for $|V_{ids}| = 0.7$. When participating users had five new identities, recall rates dropped below 10% at $|V_{ids}| = 0.5$ for all networks.

Edges sorting was also tested with a power-law distribution having $Y = 5$. These experiments resulted in a slightly higher true positive rate, which is understandable: if edges are not uniformly distributed it is more likely for an identity to have more of the original edges than the others (with higher chances to be re-identified). In another comparative experiment I modeled a variable number of new identities with power-law-like distribution with $Y \in \{2, 5\}$ and uniform edge sorting probability. Results were properly centered between cases $Y = 2$ and $Y = 5$ as the LiveJournal example shows in Fig. 5.6a.

Although by inspecting recall rates the basic model seems ineffective in impeding



(a) Recall rates

(b) Disclosure rates

Figure 5.7: Effect of edge deletion comparied to the basic model (Epinions dataset).

the attack, the disclosure rates yield better results. As shown in Fig. 5.6b, disclosure rates are significantly lower compared to recall rates[2]. From this point of view using the basic model with $Y = 5$ and uniform edge sorting probability provides strong protection for even a small ratio of applying users: the disclosure rate is at most 7.56% when $|V_{ids}| = 0.1$. By comparing the results of the two measures, we can conclude that by using the basic model it is not feasible to repel the attack, however, by using a higher number of identities the access of the attacker to information can be effectively limited.

5.2.2 Sensitivity to Edge Deletion

For testing the Nar09 against additional edge perturbation by identity separation, the realistic and best models were used with three different settings in the experiments: realistic model with minimal deletion, realistic model with random deletion, best model with random deletion (see Section 3.5.5). I executed simulations for these models with $Y = 2$, and found that recall rates strongly resemble results of the basic model (although being slightly better); thus, these methods are also incapable of repelling the attack on the network level (see Fig. 5.7a). Fortunately, disclosure rates are better compared to the basic model, e.g., results for the Epinions network are depicted in Fig. 5.7b. We can conclude that while these models are also incapable of stopping large-scale propagation, they yet perform better in privacy protection.

[2] Note: as the disclosure rate is measured for $\forall v \in \tilde{V}_{ids}$, results start from $|V_{ids}| = 0.1$.

5.3 Non-Cooperative Identity Separation

Here I analyze the limits of non-cooperative privacy-enhancing identity separation, how these effect overall results.

5.3.1 Why the Use of the Basic Model with $Y = 2$ Should be Reconsidered

While conducting the analysis, I found a case when the recall rate was notably higher for users of identity separation ($\forall v \in \tilde{V}_{ids}$) compared to the overall recall ($\forall v' \in \tilde{V}_{src}$). For low values of $|V_{ids}|$ this difference in the recall was almost constant and decreased for higher values (see values for regular seeding in Fig. 5.8).

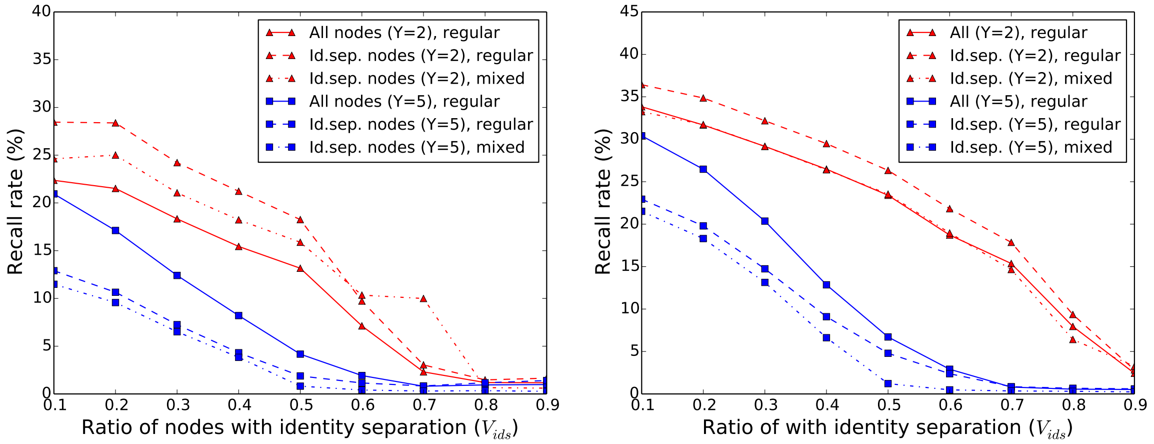
Eventually, this turned out to be caused by the seeding strategy. Throughout my experiments I used seeds that were not affected by identity separation, but after changing to mixed seeding with an equal ratio of seeds selected from $\tilde{V}_{src}$ and $\tilde{V}_{ids}$, while the overall recall rate remained unchanged, the difference disappeared for the LiveJournal and Slashdot networks, and significantly decreased for the Epinions (examples showed in Fig. 5.8). From the user perspective, the disclosure rates did not vary much by changing the seeding strategy.

This finding has an interesting impact for the attacker on choosing the seeding strategy. Using a regular seeding mechanism is a natural choice, and building fault tolerance into it against identity separation is not a trivial task. Therefore, by using the natural choice of seed identification, the attacker will also have a higher rate of correct identification for nodes protecting their privacy. However, I note that the seeding mechanism should be chosen with caution; as the analysis in Section 5.1 shows that some seeding method are not resistant to identity separation.

The core message of this finding for users aiming to protect their privacy is that they should use higher number of new identities. As examples in Fig. 5.8 shows that recall rates for users with $Y = 5$ was lower than the network average, and even if an attacker uses a mixed seeding mechanism it is also counterproductive. This advice is further strengthened by the findings of the Grasshopper algorithm. Using the basic model with $Y = 2$ against this robust attack, neither network, nor user privacy could be preserved efficiently (see Fig. 5.9a). However, using the best model with $Y = 5$ minimization of data disclosure can be achieved (see Fig. 5.9b).

5.3.2 Recall Rate Comparison for Multiple Models in Parallel

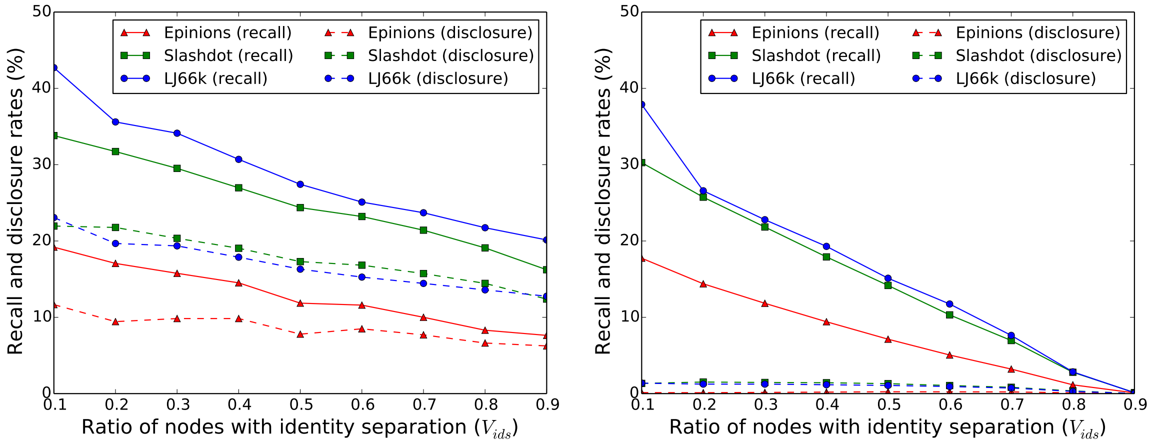
In previous experiments different identity separation models were used homogeneously. I investigated if the observed differences remain when multiple settings are



(a) Epinions network

(b) LiveJournal network

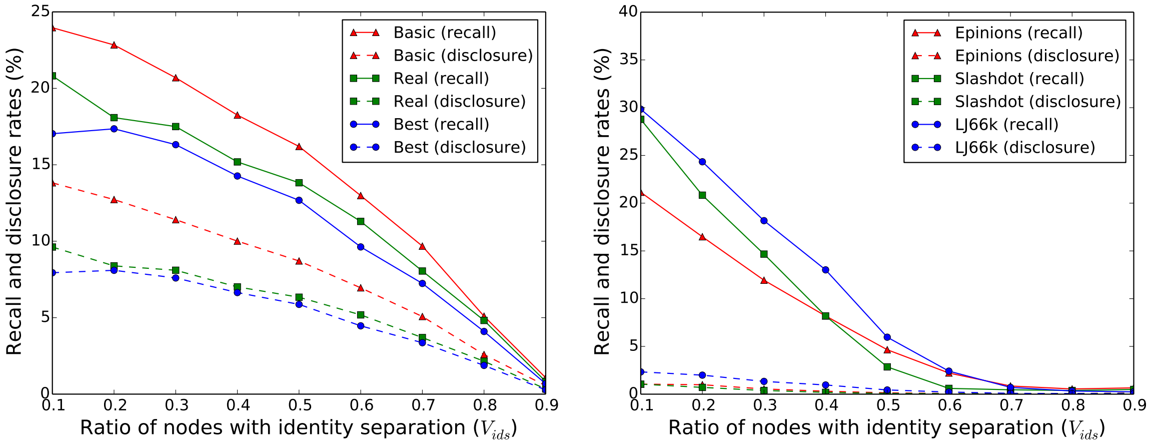
Figure 5.8: Comparison of recall rates for all nodes and ones using identity separation shows that using a low number of new identities leads is counterproductive and leads to higher recall rates than average ($Y = 2$). Therefore users should be advised to use a higher number of new identities as results suggest.



(a) Basic model, $Y = 2$

(b) Best model, $Y = 5$

Figure 5.9: The Grasshopper algorithm is quite robust against features of identity separation. For the basic model with $Y = 2$ the attack could not be defeated even with $|V_{ids}| = 0.9$. In case of the best model with $Y = 5$, Grasshopper can be defeated only with a very large fraction of participants; however, for the adopters of the technique user privacy is preserved in this case.



(a) LiveJournal network

(b) Best model with $Y = 5$

Figure 5.10: (a) When multiple models are used in parallel, users get results accordingly to the model they use. (b) Even the best model with $Y = 5$ cannot repel the attacker on a network level, just by involving the majority of users; however, privacy-enhancing users still achieve low disclosure rates.

allowed in the same network. The following models were used on the given proportion of users: basic model with uniform edge sorting probability (34% of V_{ids}), realistic model with random deletion (33% of V_{ids}), best model with random deletion (33% of V_{ids}). Results show that for the users of each setting was proportional to results measured in previous experiments, for instance, users of the best model achieved the lowest recall and disclosure rates. Simulation results in the LiveJournal graph are plotted in Fig. 5.10a for demonstration (results were measured for homogeneous groups consisting of nodes having the same setting).

5.3.3 Applying Patterns from the Twitter dataset

For measuring real-life like user behavior with identity separation-like features, I proposed two strategies to apply real-life patterns from the Twitter dataset, to see how the use of these patterns can eliminate re-identification (from dataset mentioned in Section 3.5.1). In the case of the *Twitter patterns* strategy, for a given node patterns were randomly selected from nodes with a similar degree, which determines the number of new identities and how edges need to be sorted. For an example, see Table 5.1a.

In the case of the *Twitter circle* strategy, first the number of new identities y was calculated of node v in coherence with the distribution of new identities observed in the data, where Y was limited for rational considerations (see Table 5.1b). Next, the pattern is selected randomly, with a probability proportional to its relative frequency in the dataset we used. Patterns distributed similarly as in Table 5.1a.

Simulation experiments proved these strategies to be less useful against de-anonymization. However, this is not surprising, as there is no edge deletion, but only

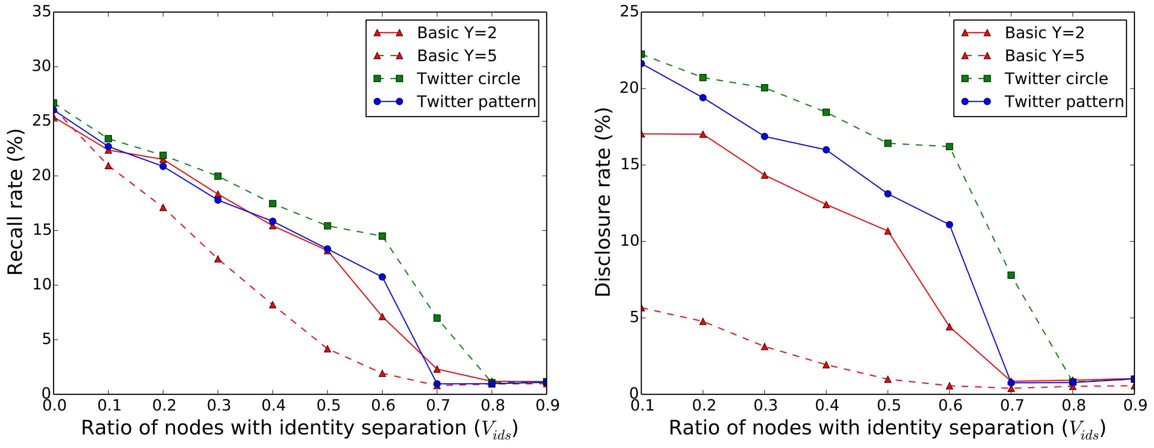
$P(X_0, X_1)$	$x_0 = 0$	$x_0 = 1$
$x_1 = 0$	0.00	0.16
$x_1 = 1$	0.82	0.02

(a) Edge sorting distribution for a node v' picked from the Twitter sample, to be used with the *Twitter patterns* strategy. The node had $deg(v') = 50$, $y_{v'} = 2$.

	$y = 2$	$y = 3$	$y = 4$	$y = 5$
$P(Y = y)$	0.49	0.27	0.14	0.08

(b) Distribution of the number of new identities applied from the Twitter dataset. Such distributions were used with the *Twitter circle* strategy.

Table 5.1: Characteristics for applying patters from the Twitter ego network dataset.



(a) Recall rates

(b) Disclosure rates

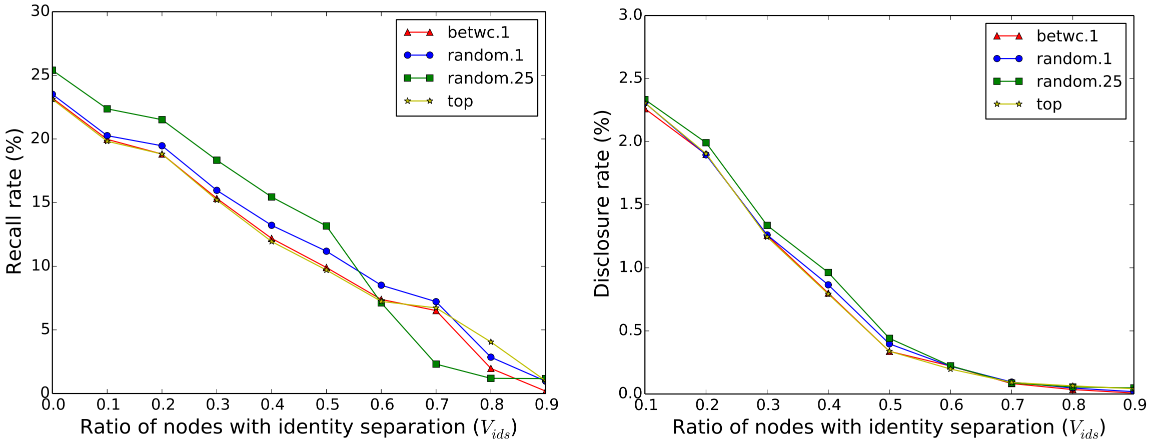
Figure 5.11: User patterns in the Twitter egonet dataset are closer to the worst model, thus also results in the Epinions dataset are worse than the basic model.

duplication, using these patterns resemble using the worst model. Examples of my results are summarized for the Epinions dataset in Fig. 5.11. In addition, it should also be considered that these results show a higher level of privacy-protection based on the Twitter dataset than expected in reality (with the same patterns). Users were reckoned to have a separated identity for each circle in the dataset, however, this is an overestimation of the use of identity separation.

5.3.4 Applying the Best Model with $Y = 5$

While none of the previously analyzed defense strategies can effectively stop the attack, identity separation can reduce disclosure rates effectively. It also turned out that increasing the number of new identities has a powerful impact on the disclosure rate, while edge perturbation has little, but notable effect. Therefore, from the user point of view, the best model with a high number of identities seems to be a quite effective setting for enhancing privacy.

I run the best model with $Y = 5$ on all test networks. Results show that even this strategy cannot prevent large-scale re-identification when only minority of users



(a) Epinions with basic model (Y=2)

(b) LJ66k with best model (Y=5)

Figure 5.12: Comparison of advanced seeding methods against `random.25`. In most cases, using other seeding measures did not lead to significantly different results.

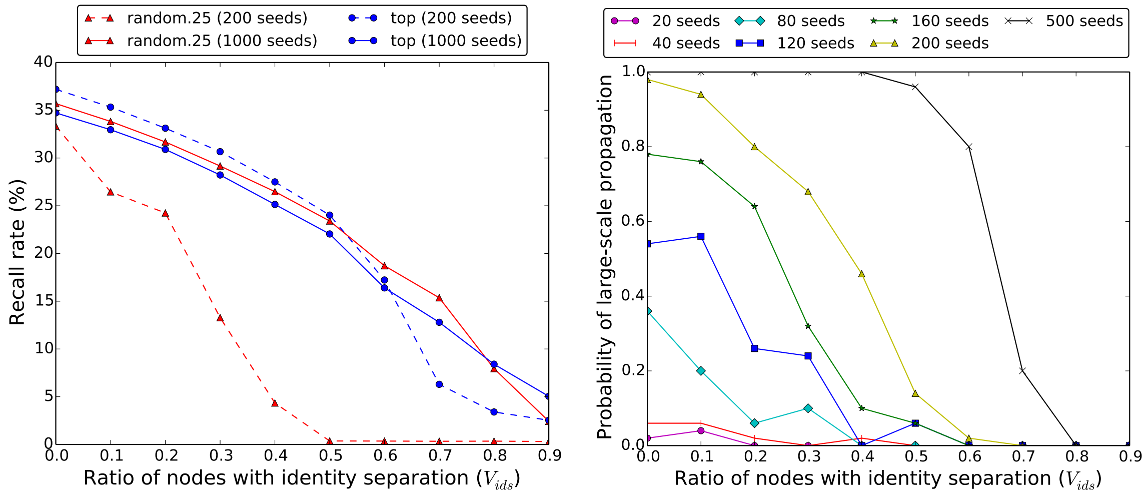
apply the technique. Instead, for all networks the re-identification rate constant monotonously decreased as $|V_{ids}|$ increased (see Fig. 5.10b). Fortunately, the setting had more convincing results for disclosure rates: even for $|V_{ids}| = 0.1$ the disclosure rates topped at 2.33%, but were typically around or under 1%. Disclosure values also continued to fall as the ratio of defending users increased.

These findings are similar to experiments with the Grasshopper algorithm, where also large adoption rates were required for protecting network privacy, although disclosure rates were equal or lower to 1.51% (see Fig. 5.9b). Results for Nar09 and Grh both suggest that using the best model with $Y = 5$ is a feasible strategy for protecting privacy individually against currently known modern attacks.

5.3.5 Using Different Seeding Methods

In coherence with the discussion of Section 4.2, the used seeding methods should also be analyzed as a part of the attacker model. Basically, I used the `random.25` method, and compared others in order to see if those are more robust against the perturbation caused by identity separation. The compared ones were the `betwc.1`, `random.1` and `top` (I refer these as the advanced seeding methods). For these only a handful of seed nodes are enough for large-scale propagation. Here I also used a constant seed size of a thousand nodes, similarly to previous experiments.

Results in the Epinions network (with basic model $Y = 2$) are show in Fig. 5.12a. Only minor differences could be observed when using different seeding methods. However, it should be noted that advanced methods seem to be a better choice when a higher ratio of users apply identity separation ($|V_{ids}| \geq 0.6$). This was also true for the recall rates in Slashdot network, and for the disclosure rates in both networks, but



(a) Recall with lower number of seeds

(b) Seeding stability with `random.25`

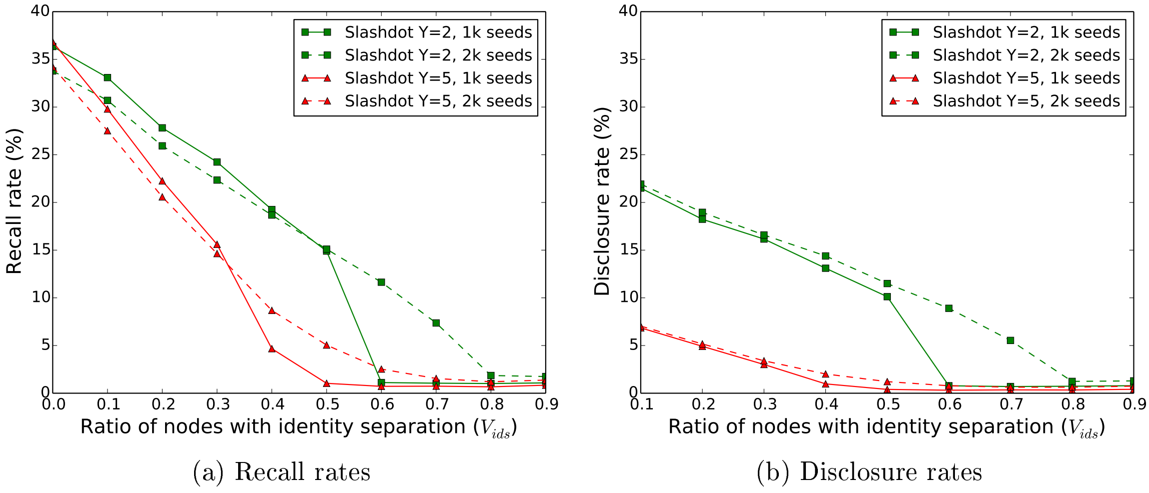
Figure 5.13: If the number of seeds is only 200 nodes (considered stable for $|V_{ids}| = 0.0$), `top` turned out to be more resistant against identity separation than `random.25` (a). Probability measurements on (b) show that for larger perturbation more seeds are needed for stable seeding, and this aspect `top` is more redundant than the other. Experiments were run on the LJ66k dataset with basic model ($Y = 2$) in both cases.

only when using the basic model with $Y = 2$.

In case when I modeled identity separation with the best model and $Y = 5$ (in any of the datasets), or if I considered the LJ66k network with either $Y = 2$ or $Y = 5$, recall and disclosure rates resembled the results shown in Fig. 5.12b. In these cases the `random.25` seeding method seemed to provide only slightly better results than the others, but essentially there were no differences.

These results alone would not justify the use of seeding methods other than `random.25`. However, when it is not possible to re-identify a large number of seeds initially, advanced methods should be considered. As Fig. 5.13a shows, the `top` method was more robust against identity separation than `random.25` when only 200 seeds were available. Results with higher seed number are also plotted for comparison, and the results are from the LJ66k dataset, using basic model with $Y = 2$. Here, `top` had enough seeds even for large $|V_{ids}|$ values, while `random.25` would need more nodes for seeding as the level of perturbation increases

This is for a simple reason: the greater the level of perturbation is in the current experiment, the more seed nodes are required to have stable seeding. In Section 4.2 I showed that the `top` selection method needs significantly less nodes for stable seeding than the `random.25` method. I demonstrate the connection between the number of seeds and stability for various $|V_{ids}|$ in Fig. 5.13b. Each experiment was run 25 times with different random seed sets for balanced results. Here, we can consider seeding to be stable, when the probability of large-scale propagation is approximately 1.0, and



(a) Recall rates

(b) Disclosure rates

Figure 5.14: The random.25 method was tested with higher seed values on the LJ66k dataset against different levels of perturbation created by the basic model with $Y = 2$ and $Y = 2$. Higher numbers of seeds led to better results when $|V_{ids}|$ was also large.

propagation can be considered large-scale when $R(\mu) \geq 0.75 \cdot R_{max}$, i.e., recall rate reaches 75% the highest observed recall for the given $|V_{ids}|$. By running measurements with identical parameters and datasets with $|V_{ids}| = 0.0$, in Section 4.2, I measured the minimum seed set size required for stable seeding ca. 80 for `top`, and ca. 180 for `random.25` for the LJ66k network. This is coherent with current measurements.

Further corollary of this finding is that another type of attackers should be also considered: an attacker can search for a seed set consisting of a low number of nodes on a trial-and-error basis until large-scale propagation appears. Even in this case, due to the design of Nar09, the error rate is likely to be low. For example, I managed to reach $R(\mu) = 27.88\%$ ($R_{max} = 36.41\%$) with only 20 nodes selected by `random.25` in the LJ66k network (and had another similar case). Considering this aspect, another type of attacker should also be considered who can successfully re-identify a large fraction of nodes even if the majority users not used identity separation.

5.3.6 Increasing the Seed Size

Another possibility of the attacker to have better results is to use larger seeds sets. Thus I tested the `random.25` seeding method with higher number of seeds (2000 nodes) on the LJ66k dataset against the basic model with $Y = 2$ and $Y = 5$. Results are shown in Fig. 5.14a and Fig. 5.14b.

Increasing the number of seed nodes only helps when the seeding method is unstable at the given perturbation rate (see Fig. 5.13b for details), e.g., when using `random.25` with only 200 seed nodes in the LJ66k network (Fig. 5.13a). However, it must be noted that a seed size of 2000 nodes is irrationally large in proportion to the overlap

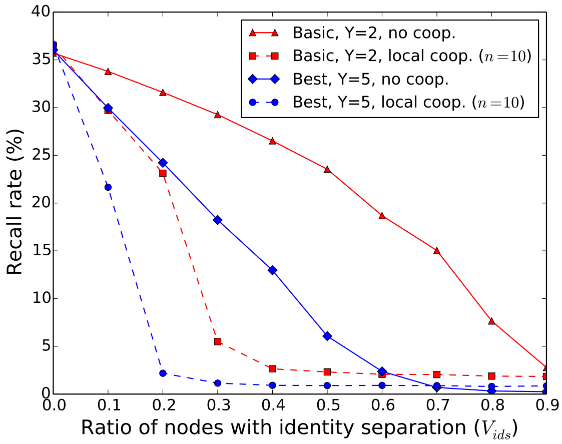


Figure 5.15: The effect of local cooperation compared to the non-cooperative settings in the LJ66k dataset.

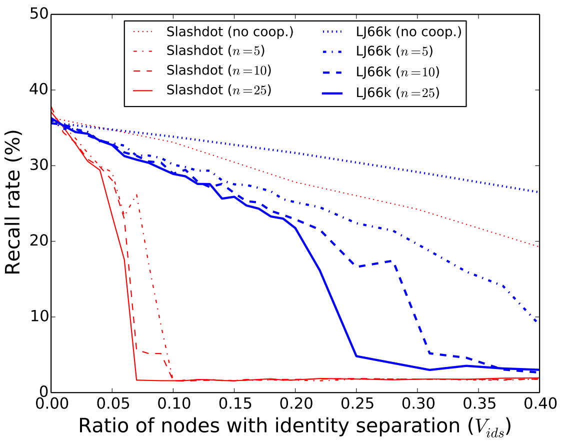


Figure 5.16: Local cooperation with different groups sizes (basic model, $Y = 2$) in two different networks. Even small groups sizes show significant difference to non-cooperation.

between the anonymized and known datasets in these experiments. For example, in the $Y = 2$ case the overlap size is just around 6000 nodes, which is only three times larger than the seed size.

5.4 Evaluation of a Local Cooperation Scheme

Previous measurements showed that non-cooperative identity separation cannot prevent the attack on the network level effectively. Therefore I investigated multiple cooperative models, focusing on the analysis of local cooperation first.

Modeling a simple local cooperation scheme including a sizing parameter n can be considered as follows. First a node is randomly selected, and then $n - 1$ nodes are sampled from its neighborhood. Initially, I expected this scheme to provide similar results as non-cooperative identity separation, due to a simple reason: the scale of the effect of such cooperation is small and limited regarding from a global point of view. Thus it should not affect the attack seriously.

Nevertheless, I evaluated this scheme for $n \in \{5, 10, 25\}$ with the basic model with $Y = 2$ and the best model with $Y = 5$, and results surprisingly showed significant progress compared to the non-cooperative case. Fig. 5.15 shows that having local cooperation with identity separation ($n = 10$) decrease the required number of participants for tackling the attack from $|V_{ids}| = 0.9$ to $|V_{ids}| = 0.3$ even with the simplest model (basic, $Y = 2$). With the best model even less participants are enough, namely $|V_{ids}| = 0.2$ (here, disclosure rates were also similar in shape to recall).

Fig. 5.16 gives finer-grained details how cooperation size n affects results. While even having small sized ($n = 5$) cooperation improves results compared to the non-cooperative setting, and after the size of the local collaboration reaches a sufficient value (here $n = 10$), further increases has a lighter effect (example given for $n = 25$).

5.5 Globally Cooperative Identity Separation

Introducing local cooperation improved results compared to the non-cooperative identity separation. In the following experiments I experimentally examined if global cooperation can provide further improvements regarding recall rates from the defending point of view.

For global cooperation, I used measures of node importance to select which nodes are reckoned to be cooperating, as these can suggest how important a node is for an attacker. In the current scenario, such a property could be compared with how re-identifiable a node statistically is within the network. Therefore I used two predictive measures on re-identification, LTA_A and LTA_{deg} measures discussed in Section 4.1.

5.5.1 Global Cooperation Based on LTA_A

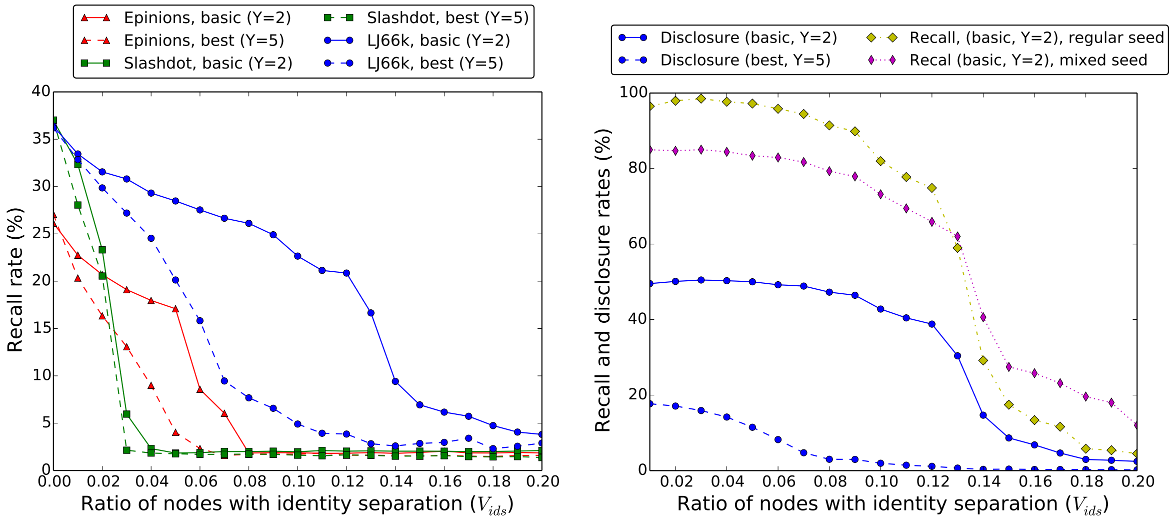
In this section I evaluate an LTA-based cooperative method that considers involving nodes that have low LTA_A values.

Analysis of LTA-based Cooperation

First, I run simulated cooperation on the basic large datasets that were used in previous experiments. In these measurements nodes using identity separation were not selected randomly, but the ones that had lowest LTA_A values. Thus $|V_{ids}| = 0.01$ means that 1% of nodes were selected to apply identity separation that had the lowest LTA scores among all nodes (this is maintained for overlapping nodes).

First, I applied this scheme with the basic model ($Y = 2$, uniform edge sorting) and also the best model ($Y = 5$, random edge deletion). Results are displayed in Fig. 5.17a. The figure shows that in this case the attack fails even for a significantly lower number of users are involved. For example, when users were selected randomly in the experiments, in the Slashdot dataset 60% of them needed to use identity separation in order to defeat the attack, while in the cooperative case only $|V_{ids}| = 4\%$ is enough (basic model). For the LJ66k network, this was as high as 90% in the non-cooperative case, while in the cooperative case for $|V_{ids}| = 15\%$ recall rates drop as $R(\mu) < 7\%$.

Disclosure rates for the LJ66k network are displayed in Fig. 5.17b; I note that disclosure rates are the quite promising here from the attacker point of view. It is clearly visible from the figure that disclosure rates are highest for the bottom LTA nodes. I also included the recall rates for these nodes, that is $R(\mu) \geq 95\%$ when



(a) Recall rates

(b) Rates on nodes using id.sep. in LJ66k

Figure 5.17: After targeting users with lowest LTA$_A$ values a significantly lower number of users is enough to stop the attack, e.g., the recall rate in the Slashdot network drops below 5% even when only $3-4\%$ of users participate (this is $50-60\%$ in the non-coordinated case). However, disclosure rates are rather high compared to the non-cooperative case: high degree nodes using identity separation are easier to be identified even despite defensive measures.

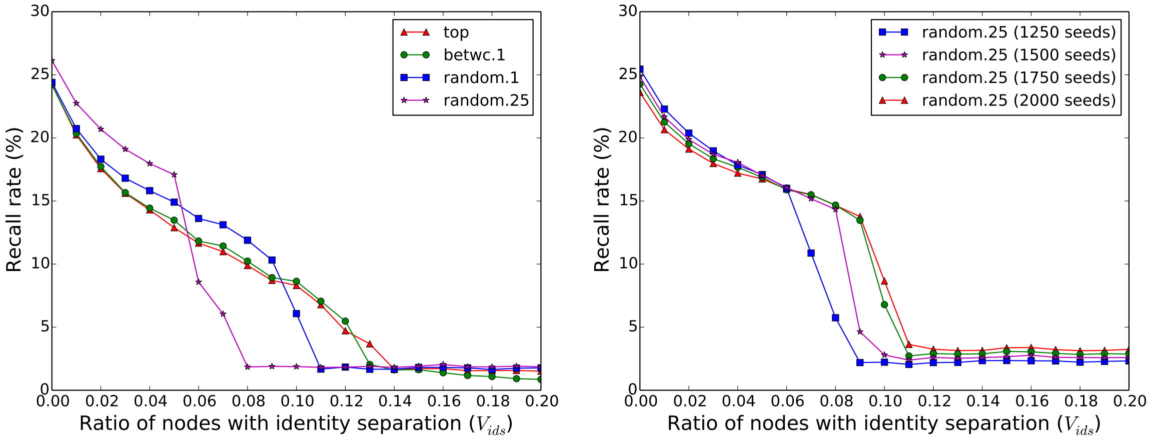
$|V_{ids}| \leq 5\%$, and remains rather high even after. This is likely because that identity separated nodes retain strong similarity with their matches in the auxiliary dataset causing one of the new identities always to be found. However, it is important to note that the disclosure rates are less important in the cooperative case, as the goal of the users here is to minimize recall rate.

In addition, the seeding method also plays a significant role in this case as well (similarly as described in Section 5.3.1). For a mixed seeding method recall rates dropped for nodes using identity separation, while still staying high due to the ease of identification (as large nodes retain a large fragment of their fingerprint even after identity separation). For these reasons discussed above, disclosure rates stayed similarly high to this case throughout the cooperative experiments.

Enhancing the Seeding Method Against Cooperative Defense

Similarly to experiments in the non-cooperative case, I compared the `random.25` seeding method to others in order to measure their robustness against identity separation. Here, the `betwc.1`, `random.1` and `top` was also used for comparison with a seed set size of a thousand nodes. Additionally, I tested `random.25` with larger seed set sizes of $1250, 1500, 1750$ and 2000.

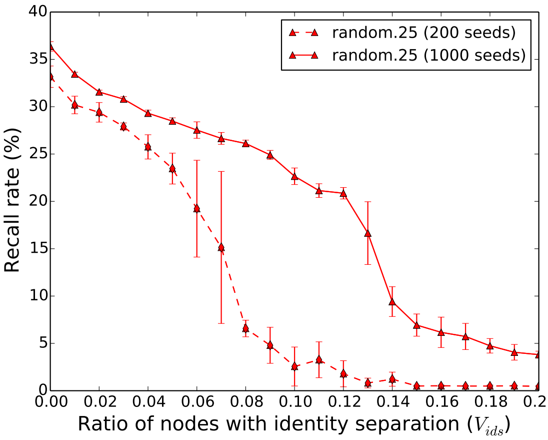
Highlighted examples of the results of these experiments are shown in Fig. 5.18; similar behavior were observed in other cases. Results indicated on the figures clearly show that the attacker has only a little control over the overall results: neither using



(a) Using different seeding methods

(b) `Random.25` with increased seed set

Figure 5.18: The attacker has little control over results. Neither using other seeding methods (a), nor increasing the seed set size (b) can significantly improve recall rates. (examples from the Epinions dataset, using LTA$_A$ based global cooperation)



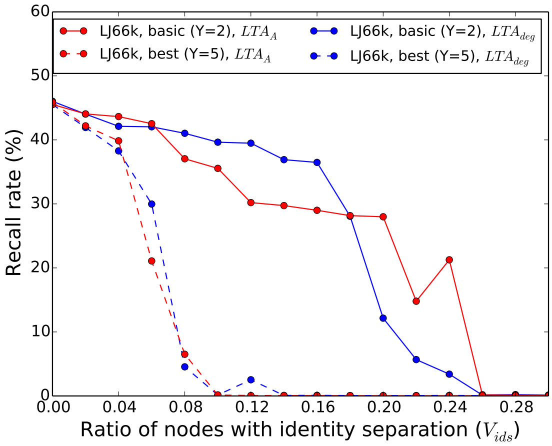
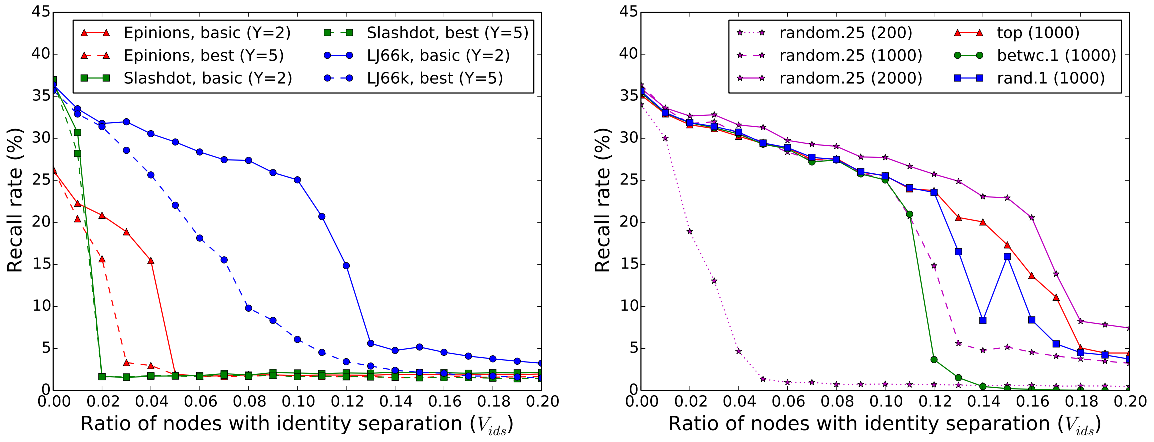


Figure 5.19: Experiments with low seed numbers; recall dependency on seed set size is displayed in the LJ66k dataset. These results also show that lower seed numbers cause greater instability, too (using LTA$_A$ based global cooperation).

Figure 5.20: Global cooperation can significantly decrease the minimum number of adopting users required also for defeating the Grasshopper attack.



(a) Degree-based global cooperation in all test networks. (Nar09 algorithm was initialized with 1000 `random.25` seeds.) A significant drop in the number of required participants is visible in results.

(b) Comparison of more advanced seeding methods and different seed sizes for the `random.25` method (results are from the LJ66k network with basic model, $y = 2$).

Figure 5.21: Re-identification recall rates in various settings when participants cooperate according to the degree-based global cooperation scheme.

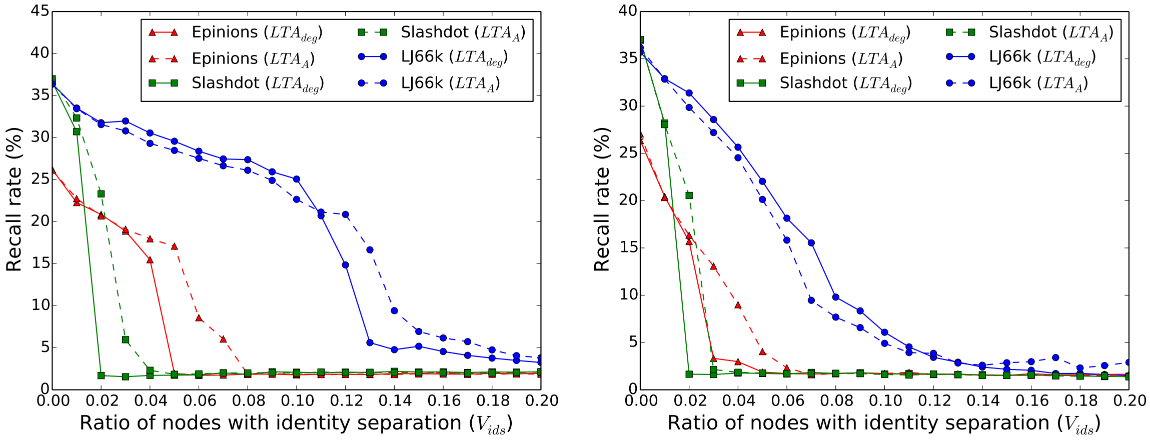
different seeding methods (Fig. 5.18a), nor increasing the seed set size could improve recall rates significantly (Fig. 5.18b).

The drop in the recall rate of `random.25` after $|V_{ids}| = 0.05$ in Fig. 5.18a is also caused by the sensitivity to the number of seeds, which is further detailed with an example in Fig. 5.19. In this cooperative case the seed stability also depends on the number of seeds, similarly to the case of 200 seeds in Fig. 5.13b: with a higher number of seeds, large-scale propagation can be achieved with a higher probability for each perturbation settings (i.e., different values of $|V_{ids}|$). Thus, we can conclude that this seed size dependency causes this minor difference between the results measured.

5.5.2 Degree-Based Global Cooperation

I also implemented global cooperation as having the top nodes committing identity separation according to the given model (i.e., top $|V_{ids}|$ ratio of the network). Main results displayed in Fig. 5.21a show that global cooperation is quite effective method compared to the non-cooperative setting, local cooperation (comparison with LTA_A is discussed in the following section). For example, in the case of the Slashdot network only 2% of the whole network needs to participate in order to stop the attack, which is $50 - 60\%$ in the non-cooperative case, and $20 - 30\%$ when local cooperation is applied. Results in other datasets show similarly improved results.

Simulation were also run to see if a stronger attacker with enhanced seeding can be more robust against identity separation, thus being able to achieve higher recall rates



(a) Basic model, $Y = 2$.

(b) Best model, $Y = 5$.

Figure 5.22: Comparison of results between cooperation organized by LTA_A and LTA_{deg}. Dashed lines represent results for LTA_A, and solid ones are for LTA_{deg}.

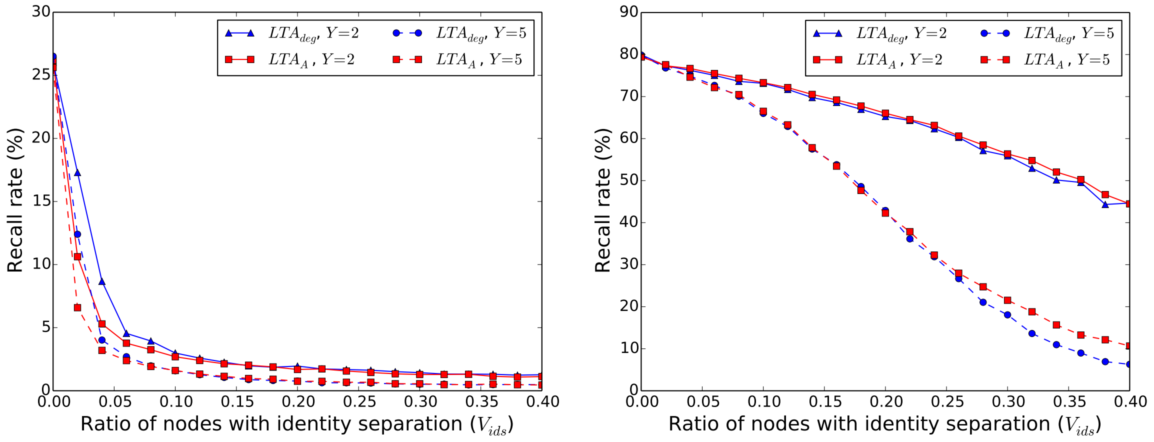
(for results see Fig. 5.21b). According to the simulations, even an attacker having a higher quality seed set or with a larger one cannot commit a significantly more robust attack, as the required minimum of participants only increase from 13% (`random.25` seeding method) to 18% (`top`) in the LJ66k network. Experiments also showed that if the attacker has less seed nodes, much lower participation rates can cause the attack to fail. For example, if the attacker only has 200 seed nodes with `random.25` $|V_{ids}| = 0.04$ is enough for forcing low recall rates as $R(\mu) \leq 5\%$.

5.5.3 Comparison of Degree- and LTA-based Schemes

For the comparison of the results of LTA_A and LTA_{deg} organized cooperation I plotted the results on the same figures as shown in Fig. 5.22. For first sight, the figure shows that LTA_{deg} is slightly better in general, but interestingly this is not the case for the LJ66k dataset, where LTA_A also had better correlation rates (see Section 4.1.2 for more details). We could think this might be true in other networks, but interestingly this is not the case.

Accordingly to Fig. 5.23, LTA_A could produce better results in the DBLP80k dataset, but not in the FB30k dataset (in the PKC30k dataset none had obviously better results). However, results in the FB30k network, has two interesting characteristics showed in Fig. 5.23b. First, recall rate was initially as high as $R(\mu) \simeq 79.5\%$, which is more than the double than in other networks (second highest value was measured in the PKC30k network with $R(\mu) \simeq 48\%$). Second, this network proved to be rather robust against identity separation, as even global cooperation with $Y = 5$ could hardly decrease recall rates even with $|V_{ids}| = 0.4$.

In simulation experiments of the Grasshopper algorithm I checked if the minimum



(a) Results from the DBLP80k network.

(b) Results from the FB30k network.

Figure 5.23: Comparison of results between cooperation organized by LTA$_A$ and LTA$_{deg}$ in further datasets. The advantage of LTA$_A$ in networks having a similar degree distribution to LJ66k cannot be stated generally.

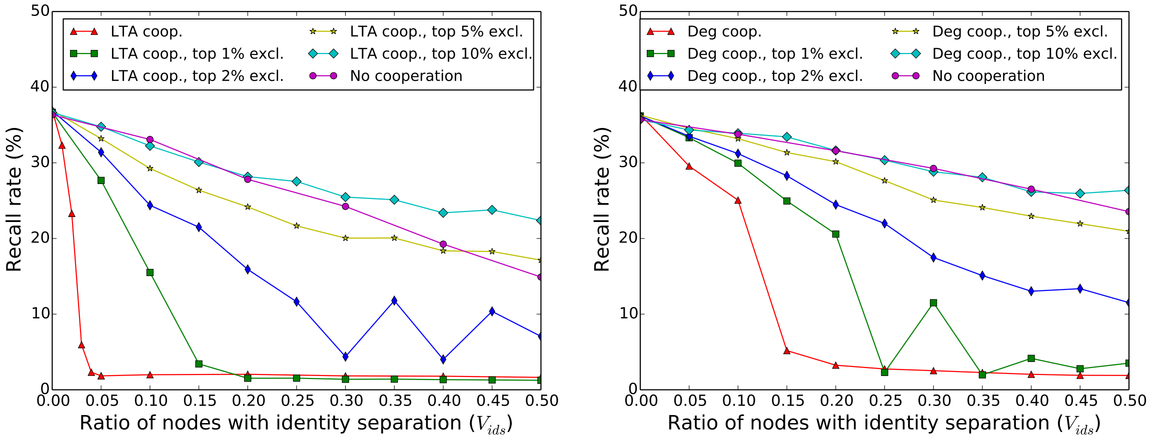
adoption rate can be decreased with global cooperation (see Fig. 5.20). Compared to the results of the non-cooperative case, we can observe progress, and the Grh attack proved to be more robust than Nar09. However, differences between results of LTA$_A$ and LTA$_{deg}$ are not outstanding, thus we can conclude that each method is feasible for tackling the Grasshopper attack by using a globally cooperative strategy (results of LTA$_A$ is only subtly better).

5.6 Characterizing the Importance of Top Nodes

Cases analyzed until this point are based on the assumption that all users cooperate to stop the attack. However, in a real life scenario it is likely that only a subset of the selected users would participate. Furthermore, the high degree nodes are the ones that are more likely to refuse cooperation, e.g., because such users do not want to divide their audience. On the contrary, we could expect that these users to use less visible solutions, such as decoys to hide their more privacy-sensitive activities.

Thus I analyzed how it affects the overall results if a given percent of the top degree nodes do not cooperate with others. Results are shown in Fig. 5.24 for both global cooperation strategies. Compared to when all users participate, it turns out that even if only 1% of top degree users denies cooperation a significantly larger ratio of users need to be involved for successfully tackling the attack.

Comparing the results of non-cooperative behavior to the cases when a large fraction of top nodes avoid (global) cooperation, the importance of top degree nodes becomes clear. For example, in case of LTA$_A$ driven global cooperation (see Fig. 5.24a), when 5%, 10% of top degree users are excluded from identity separation, overall results get



(a) Slashdot network and LTA$_A$ cooperation.

(b) LJ66k network and LTA$_{deg}$ cooperation.

Figure 5.24: Results of globally cooperative identity separation when different ratio of top degree nodes are excluded (basic model with $Y = 2$ in both cases). Even slight slight absence of top degree nodes favors heavily the attacker.

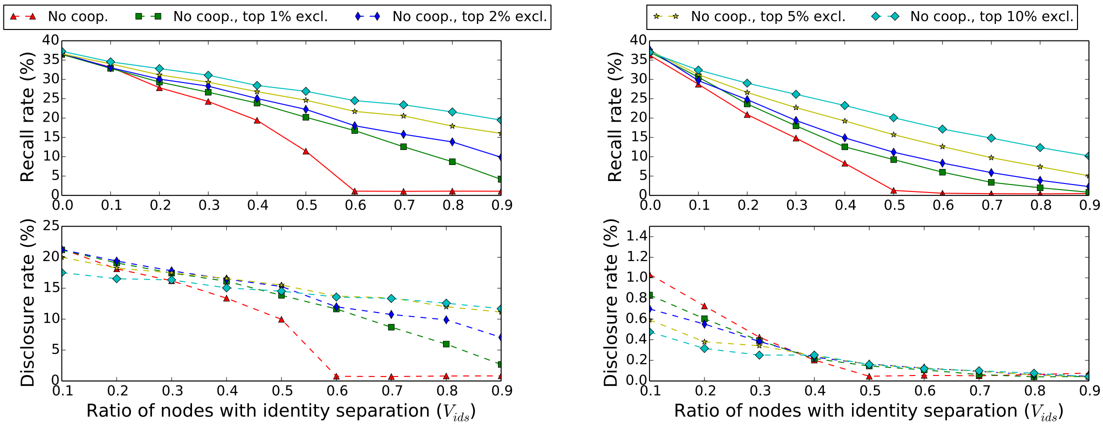
quite close to the non-cooperative case. Furthermore, for $|V_{ids}| \geq 0.4$ recall rates rise above the non-cooperative case.

Results are similar for degree based cooperation (see Fig. 5.24b), as this scheme reacts also sensitively even to loosing the top 1% of participants. If top 10% rejects cooperation, results are roughly equal compared to the non-cooperative setting. Results get even worse than the non-cooperative case after $|V_{ids}|$ reaches $0.40 - 0.45$.

Due to these findings, I revisited the non-cooperative setting by running measurements when top nodes are excluded similarly. Results are shown in Fig. 5.25, detailing both recall and disclosure rates in the Slashdot network. The figure clearly shows that the commitment of top nodes is also essential when there is no cooperation. In this case I also provided details on disclosure rates, and for the best model these rates stay rather low regardless of the proportion of top users refusing participation.

5.7 Conclusion

In Section 5.1 I provided examples for analyzing the failure probability of global identification of nodes, which is usually used for seeding (or initializing) the propagation phase of algorithms such as Nar09. In particular, I have shown how failure probability can be estimated for two seeding methods. I provided suitable strategies that could be adopted against k-top seeding which concerns high degree nodes (e.g., basic model $Y = 2$ with $p_1 = 0.5$), and also a flexible strategy set could be applied for tackling clique based seed identification (e.g., $0.2 \leq p_1, p_2 \leq 0.8$ with the basic model and $Y = 2$). Similarly to these results, models proposed in Section 3.5 can be used for the evaluation of other seeding and node re-identification methods, also.



(a) Basic model, $Y = 2$, uniform edge sorting.

(b) Best model, $Y = 5$, random deletion.

Figure 5.25: Recall and disclosure rates in the Slashdot network with the non-cooperative setting. When top degree nodes do not participate, results significantly decline; however, using $Y = 5$ pays off even in those cases, as the disclosure rates are very low, around $0.4 - 1.0\%$ (see Disclosure rates on (b)).

Subsequently in Sections 5.2 - 5.6 I analyzed the effect of identity separation on the propagation phase. Non-cooperative identity separation turned out to be ineffective in tackling the attack on the network level, while from a personal point of view the best model (with random deletion and $Y = 5$) showed to be effective in hiding personal information. I furthermore analyzed several cooperative methods, which turned to be effective in tackling the attack even when almost a small group of users adopt identity separation. However, I concluded in Section 5.6 that cooperation only works efficiently when top nodes also adopt the technique.

This finding leads to a notable conclusion over all results in this chapter, i.e., identity separation depends on a strong constraint to defeat structural re-identification successfully. From the user point of view this means that the best strategy to seek individual privacy protection (e.g., using the best model with a large number of identities) as high degree nodes may not participate. This is also for the benefit for the whole network, as the more users adopt the technique the more likely that de-anonymization would fail over the network. Due to this finding, in the following work I focused on researching individual strategies that enable a higher level of information hiding.

Chapter 6

Evaluation of Individual Strategies

> ’All human beings have three lives: public, private, and secret.’ (Gabriel García Márquez)

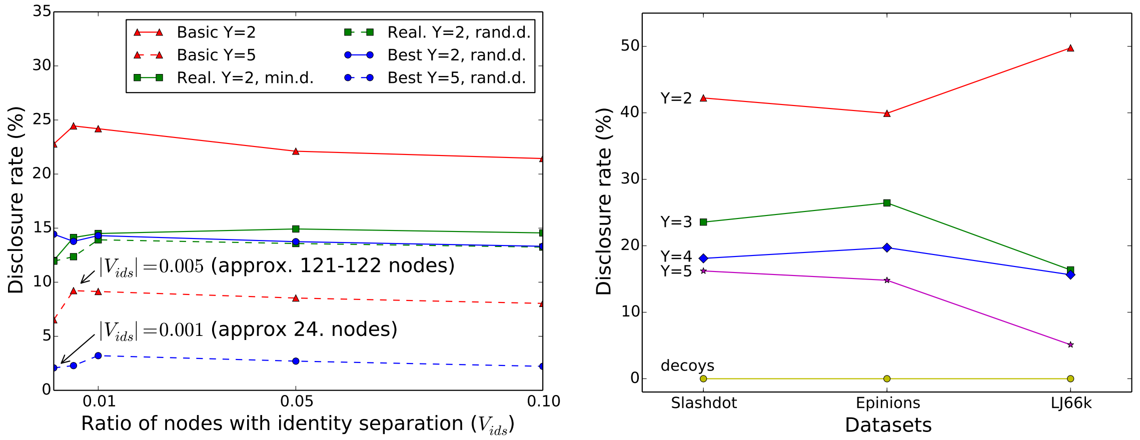
Previously I have shown that it is hard to defeat the attack on a network level. Therefore, for a single user, it is more desirable to focus on preserving individual privacy, even if network privacy is breached. In this chapter, I analyze individually applicable strategies that require no cooperation, but can be expected to provide strong privacy.

In Section 6.1 I analyze if a small number of participants (or even theoretically a single person) can use identity separation to achieve disclosure rates similar to the cases when a larger fraction of users adopted the technique. Then, in Section 6.2, a decoying technique is introduced that allows for selectively hiding information from the prying eye of an adversary, as aforementioned methods did not allow determining which identity should be perceived more sensitive than others. In Section 6.3, I show an example how Nar09 can be modified to find even multiple identities, and with simulations of this modified attack I characterize lower estimates for the finding probabilities of identities.

In Section 6.4 I provide analysis of techniques which can have theoretical guarantees on privacy. One of these techniques is based on a k-anonymity variant fitting the current context, and the other is a novel technique proposed first in our work in [J1]. Finally, I conclude in Section 6.5.

6.1 Can Small Groups and Individual Users Protect Their Privacy?

When looking for individual privacy-enhancing strategies for identity management, first we need to know if a small users or a single user can use identity separation to preserve privacy. In case of the current measurements, this means that it should



(a) Low participation rates in LJ66k

(b) Single user with best model or using decoy

Figure 6.1: In the search of the most effective privacy-enhancing strategies when applied by a few. Recall and disclosure rates in the best model is quite competitive, even when only a handful of users apply identity separation in such a way (left), but also if only a single users protects his privacy that way (right).

be tested that if a node applies identity separation then disclosure rates should stay low. Therefore, I also examined disclosure rates for cases when participation rates $|V_{ids}|$ were low such as 1‰ of V_{tar}, meaning only a few tens or around a hundred of users using identity separation from $\tilde{V}_{tar}$. As seen in Fig. 6.1a, experiments resulted in approximately constant disclosure rates for all models. Due to the low number of users, simulations were run 15 times on each dataset; yet visible variability for $|V_{ids}| < 0.01$ is likely to be due to the small sample sizes of nodes.

Furthermore, I analyzed the case when only a single user uses identity separation (best model with random deletion) or a decoy identity. Within these experiments 20 different perturbed datasets were created in which only a single user was chosen to be using privacy protection measures (with the constraint of $\deg(v_i) \geq 30$). The attack algorithm were run with 15 different seed sets on each. (N.b. this is only a hypothetical measurement, as within these cases the difference delta is only a single node, this could be found very efficiently if the public identity of the node is known in G_{src}.)

Results are summarized in Fig. 6.1b. For the nodes using identity separation the disclosure rate was somewhat proportional with the number of identities used in all networks (decoy related experiments are discussed in the following Section 6.2). Therefore we can conclude that even if only a few users use the best model with $Y = 5$, their privacy is protected as the attacker can reveal only a few percent of sensitive information, but for controlled information hiding the use of the decoy model is advised, which provided the best results.

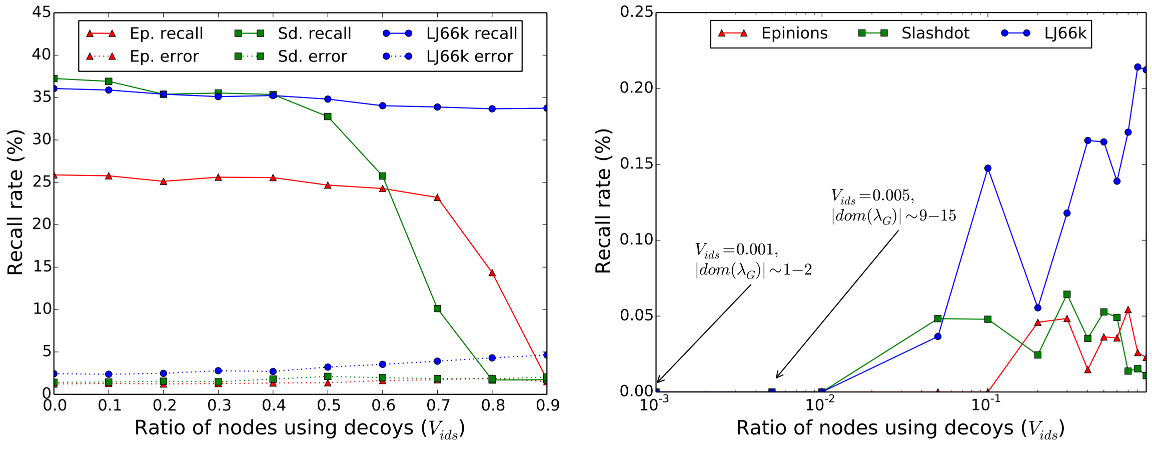
6.2 Using Decoys: Placing the User in the Decision-Making Position

Strategies discussed so far work on statistical basis, and lack user control: the user cannot decide what he wishes to hide from the attacker. Regarding an attacker capable of achieving large-scale re-identification this limits the possibilities of the user in protecting his data. For evaluating a simple scheme that put the user into the position of control, I propose a simple method by utilizing decoy identities. Methods used in real-life situations can be adapted to hypothetical attacker strategies and to the type of information for hiding, e.g., using structural steganography for hiding nodes [13]; however, in forthcoming section I propose more sophisticated strategies.

The decoy strategy on nodes $v_i \in \tilde{V}_{tar}$ were applied if $\deg(v_i) \geq 30$. This criteria resulted in having a significantly smaller set of applicable nodes, e.g., in LJ66k, even for $|V_{ids}| = 0.9$ meant only $\sim 11.2\%$ of $\tilde{V}_{tar}$. In order to apply the decoy strategy, first we need to create a decoy node v_i^P (public profile) representing non-sensitive connections with the goal of capturing the attention of attacker algorithm. Node v_i^P is assigned 90% of the acquaintances v_i has. Next, a hidden node v_i^H is created having the rest 10% of neighbors for modeling sensitive relationships, and an additional 10% that overlaps with the neighbors of v_i^P.

Simulations with this method were run with 15 times on each generated perturbed dataset. The decoy method showed promising results. From the attacker point of view the algorithm achieved misleadingly high recall rates until large number of decoys appeared (see Fig. 6.2a), while error rates constantly stayed lower than 5%. From the user perspective, privacy-protecting nodes achieved of revealing little sensitive information as shown in Fig. 6.2b, which is even lower than using the best model with $Y = 5$ (compare results with Section 6.1 or 5.3.4). Recall rates were typically small for hidden nodes, less than 0.25% within all test networks. The visible variability is negligible, and likely due to small sample sizes, as only a few nodes used the decoy method.

This simple method can be defeated when the attacker optimizes for this specific user strategy. For instance the attacker may create a new algorithm that is always able to discover both v_i^P and v_i^H, or at least one of them. In that case, given the background knowledge, the attacker then may be able to distinguish between the discovered partial identity nodes, and as a result, able to derive conclusions regarding the sensitive attributes. This can be done by decreasing the certainty of the sensitive attribute; methods such as k-anonymity can do that.



(a) Recall sensitivity to the use of decoys

(b) Recall rate for hidden nodes

Figure 6.2: The use of decoy nodes only affected de-anonymization when it was used in large-scales, and only a tiny fraction of hidden nodes was re-identified.

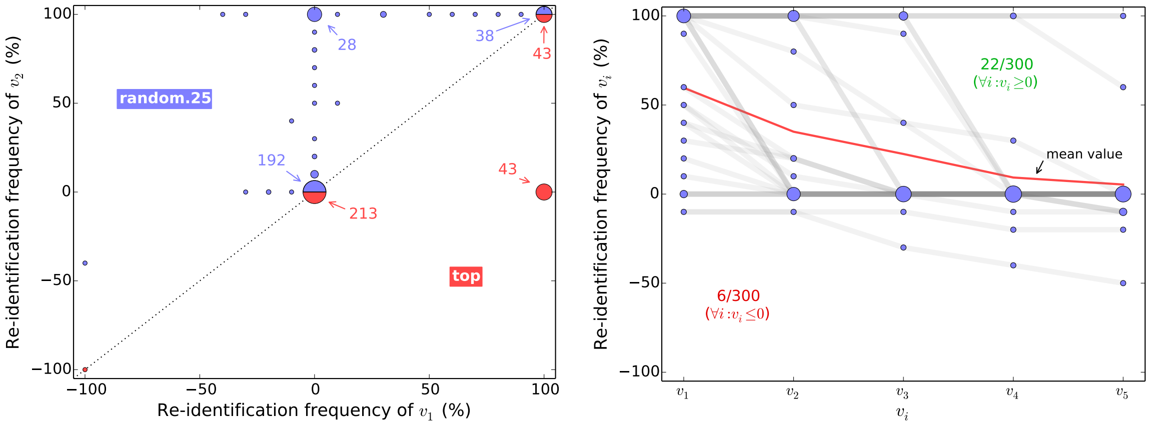
6.3 Measuring the Probability of Node Discovery and Reversibility of Identity Separation

One of the drawback of the Nar09 algorithm, that it is not designed to find multiple identities. Related to this question, it would be interesting to see how hard it is to recover multiple identities which would significantly increase disclosure rates. In order to resolve these issues, I present a method for estimating the discovery probabilities of nodes. This works with a slightly modified version of Nar09 that provides lower estimates of discovery probabilities (as other algorithm might work more accurately).

The drawback of Nar09 is that it can only assign a single identity of $v_{n\backslash i} \in G_{tar}$ to $v_n \in G_{src}$ as a match, and according to my measurements, the algorithm is quite deterministic in this: if it gives $\mu(v_n) = v_{n\backslash i}$ once, then it will yield the same match with high probability in subsequent runs (see Fig. 3.3 in Section 3.4); thus we would not have any information on the finding probability of other identities.

In order to circumvent this problem, I committed the following modification to Nar09. For a given node v_n^{tar}, measurements were run iteratively for $\forall v_{n\backslash i} \in \lambda_G(v_n)$. In each round $\forall j \neq i : v_{n\backslash j}$ were removed, and then Nar09 were run 10 times. Node discovery score $S(v_{n\backslash i})$ are monitored through the experiment for each separated identity. 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 the experiments, these measurements were run on perturbed datasets of two types, that were derived for all three networks (resulting in six datasets). In the first case I applied the basic model with $Y = 2$ (uniform edge sorting), and in the second



(a) For the case of using two identities ($Y = 2$), re-identification frequency was measured 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) Seeding method `random.25` was used 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.

Figure 6.3: 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.

case I applied the best model with $Y = 5$ (random deletion) for $|V_{ids}| = 0.1$. Next, I randomly selected 100 nodes from all six datasets having exactly $Y = 2$ or $Y = 5$, and run the aforementioned simulations regarding the selected nodes. The results are summarized in Fig. 6.3.

For testing the effect of seeding also, I used `random.25` and the `top` seeding methods for the re-identification of users having two identities. Fig. 6.3a 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 is 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. In the current experiments, no mixed cases could be observed, where some identities were correctly, and others were falsely identified several times (i.e., for $v_n^{tar} : \exists v_{n\backslash i}, v_{n\backslash j}$ that $S(v_{n\backslash i}) < 0, S(v_{n\backslash j}) > 0$). These results shed further light on the reason behind why identity separation with 5 identities produced good results in previous measurements: these cases have very low re-identification rates and even if there is correct one, only a fraction 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 ($S(v_{n\setminus i}) = 10$).

These results indicate that using five identities is strong enough against naive attackers (using Nar09). However, this is problematic: the user rarely knows the whole network, and results can also depend on the used seeding method, which cannot be certainly known apriori to the attack. Thus, it would be rational to limit the required user knowledge to a two-hop neighborhood for the measurements, but unfortunately, using only such a limited knowledge, I managed to succeeded in approximating these probabilities only in small networks (e.g., few thousand nodes), which cannot be considered lifelike. Fortunately, one does not necessarily need to know these probabilities in order to have significant protection (such as the ones proposed in the subsequent sections).

These measurements are also interesting from an adversarial point of view, too. Theoretically the attacker can also work along the modified version presented here: run the modified version of Nar09 once, then 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 my measurements (shown in Fig. 6.3), with Nar09, this can be done only to a very small fragment of the nodes using identity separation, but this finding can open an interesting line of future work.

6.4 Advanced Strategies for Protecting Individual Privacy

Using the best model or even the proposed simple decoying scheme can statistically provide some privacy protection, but have no guarantees, e.g., a stronger attacker in the future using a more advanced algorithm or obtaining a better background knowledge could find the proper identity regardless of user efforts. In previous sections I have also shown that cooperative identity separation techniques are efficient from the network point of view, but require the cooperation of high degree nodes, which cannot be guaranteed, and likely to fail in many cases.

Therefore, the need emerges for analyzing non-cooperative techniques being able to provide privacy guarantees on an individual level. The first technique I discuss is k-anonymity [36], 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 his identities to its neighbors for hiding the assigned sensitive attribute in an anonymity set size of k. I

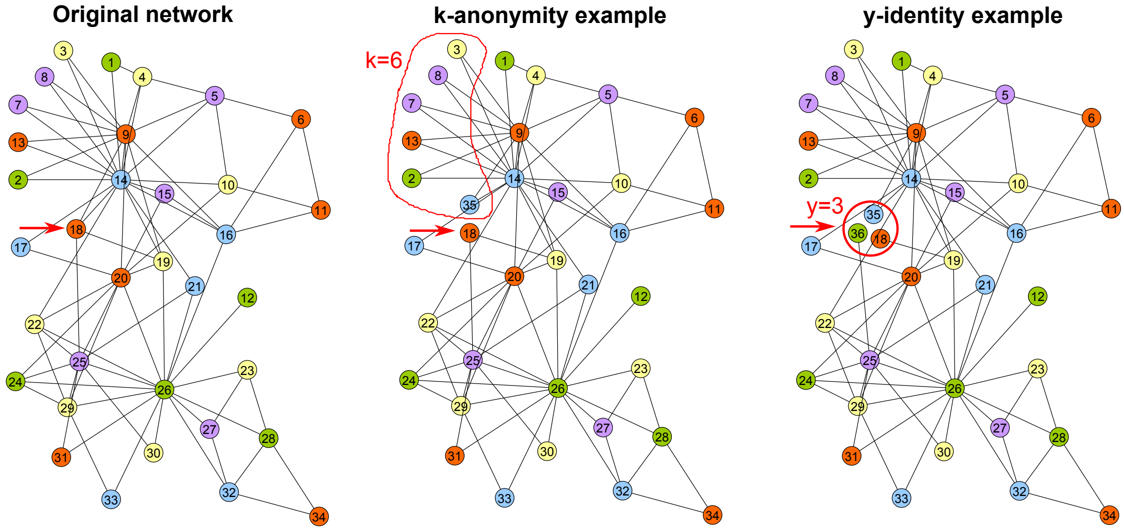


Figure 6.4: The k-anonymity and the y-identity models illustrated by examples on the karate network [71]. Colors represent privacy sensitive values published in the sanitized network.

propose a novel technique called y-identity that is based on the idea that the user can create y new identities and hide the sensitive information in one of those randomly, resulting in an anonymity set size of y (analogously to the parameter k in k-anonymity).

These two approaches are presented in Fig. 6.4 on the karate network [71]. The original karate club network is shown on the left, resembling a sanitized network, having the colors representing hypothetical sensitive attributes of nodes (which are otherwise inaccessible). In case of using identity separation with k-anonymity (middle of Fig. 6.4), the sensitive attribute of node v_{18} assigned to new identity v_{35} is now protected with $P(S = \text{'blue'}) = \frac{1}{6}$, as there 5 other nodes with the same structural fingerprint $f_{\text{Nar09}} = \{v_9, v_{14}\}$. For using the y-identity model (right of Fig. 6.4), if the attacker can even reveal all partial identities related to v_{18} (which are v_{18}, v_{35}, v_{36}), the sensitive attribute can be guessed with $P(S = \text{'blue'}) = \frac{1}{3}$. Please also note that here y-identity is used in combination with k-anonymity, as node v_{35} is also part of an anonymity set.

While the subtle differences are highly visible in this example, this should not be the case in real-life scenarios, as the background knowledge of the attacker should strongly differ from the sanitized datasets. In the case of the y-identity model, even if the attacker could reveal the fact that $\lambda_G(v_{18}) \rightrightarrows \{v_{18}, v_{35}, v_{36}\}$, the attacker could only learn a distribution $P_{18}(S)$ of the sensitive attribute. This distribution can be harmonized with $P(S)$, the distribution over the whole network, to have the attacker learn nothing by this discovery.

6.4.1 Evaluation of k-anonymity

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 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 uniquely identified with the quasi-identifier of `{5-digit ZIP, gender, date of birth}` [72].

Definition 3. *k-anonymity. A dataset is k-anonymous if for all entries there are at least* k-1 *other entries with the same quasi-identifiers [36].*

Despite it has been shown that the concept of k-anonymity is inappropriate for anonymizing data with high dimensionality [73], it is applied and analyzed in many contexts even for the sanitization of social network structural data [74]. There are also known weaknesses of k-anonymity, for example that despite of anonymization the attacker can still learn information as the distribution of the sensitive attributes in the k-anonymous groups can significantly deviate from the global distribution. Subsequent models aim to patch this vulnerability, such as l-diversity and t-closeness [75].

I find using overall network anonymization methods unrealistic as they require consent and interaction on the behalf of the service provider. Due to this reason, I analyze a method for applying k-anonymity individually to tackle structural re-identification attacks (which might be later replaced with advanced methods like t-closeness). As discussed earlier, re-identification algorithms such as Nar09 compare nodes to their friends-of-friends (the 2-hop neighborhood), and therefore the concept of k-anonymity to can be extended to identity separation if the quasi identifier, or the fingerprint function $f_{\mathrm{Nar09}}(\cdot)$, of a node is based on his neighbors.

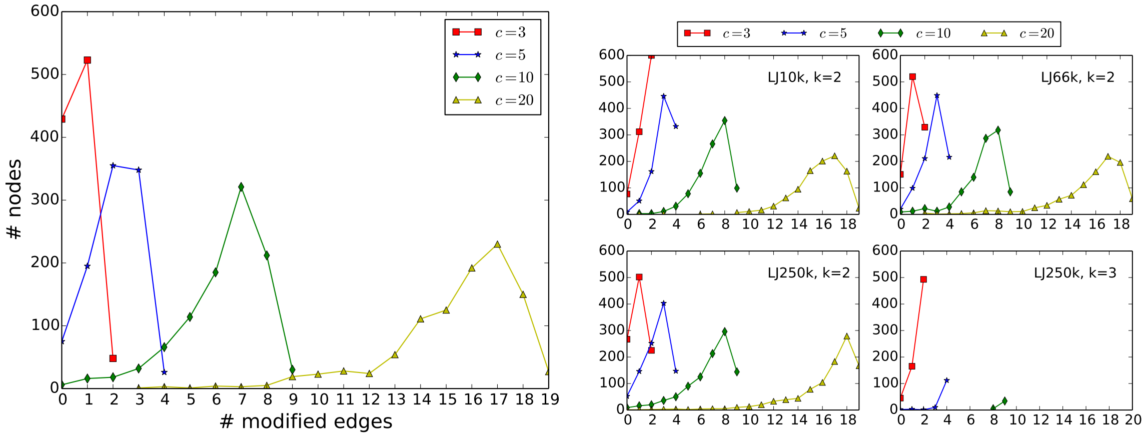
Definition 4. $(k,2)$*-anonymity. A user* $v_n \in G$ *is* $(k,2)$*-anonymous if there are at least* k-1 *other (non-adjacent) users having exactly the same neighborhood, i.e.,*

$$\exists A_k = \{v_i : \forall v_i \in V_n^2, V_i = V_n\} \to |A| = k,$$

where V_i *denotes the neighbor set of* v_i*, and* V_i^2 *denotes the neighbors-of-neighbors of* v_i*.*

This definition can be extended to allow edges between the members of the anonymity set A_k. In this case only the edges going out of A_k need to be identical, and the internal structure need to be a subgraph that is a symmetric graph. Before extending the definition, first my goal was to test if the simpler definition works.

Therefore I have constructed an algorithm (Alg. 1), called `K-AnonymizeNode`, for finding $(k,2)$-anonymous settings for users planning to apply identity separation. The



(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 develops into an even worse scenario.

Figure 6.5: $(k, 2)$-anonymity with edge modification in action. Results shows this method is not feasible as an individual privacy-enhancing strategy due to the great diversity in network structure.

algorithm assumed to know the network structure in a 2-hop distance; in the description I noted this knowledge as graph G. 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 seeks alternatives where new friendships need to be created in order to meet the criteria of $(k, 2)$-anonymity.

With `K-AnonymizeNode`, I 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 my experiments are shown in Fig. 6.5. I selected results from Epinions dataset with $k = 2$ for explanation in Fig. 6.5a. 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. Similar results can be observed in other networks, and also when analyzing whether this property differ as the network size change – see Fig. 6.5b. For greater (and practical) values of k achieving anonymity required adding even more edges if at all it was possible to reach.

Therefore, I concluded that $(k, 2)$-anonymity is not a valid option for individually protecting privacy, as the structure of social networks is not making such techniques feasible. As a continuation of the research, I analyzed an alternative method called y-identity. However, this is a not a surprising result. It has been previously shown that in general, k-anonymity fails when there is a high dimensionality in the data [73]. This is obviously true for social networks, and has been the feature exploited in the line of de-anonymization attacks capable of large-scale re-identification [2, 76]. The current finding reveals that individually using this technique is also a hard problem.

Algorithm 1: $(k, 2)$-anonymity with edge modification. It takes as input: the graph structure G, a node v_i selected for identity separation, c denoting the number of connections to anonymize, and parameter k of k-anonymity.

```
procedure K-ANONYMIZENODE($G, v_i, c, k$)
    Calculate $V_i, V_i^2$
    $c' \leftarrow c$, $V_k \leftarrow \{\}$, $E_k \leftarrow \{\}$
    while $c' \geq 1$ and $|V_k| = 0$ do
        $\kappa \leftarrow \{\}$                              ▷ Groups having $c'$ common neighbors with $v_i$
        for all $v_j \in V_i^2$ do
            $V_{i\cap j} \leftarrow V_i \cap G.nbrs(v_j)$
            if $|V_j| = c$ and $|V_{i\cap j}| = c'$ then
                $\kappa[V_{i\cap j}] \leftarrow \kappa[V_{i\cap j}] \cup \{v_j\}$
            end if
        end for
        for all $\kappa[V_{i\cap j}]$ if $|\kappa[V_{i\cap j}]| \geq k - 1$ do
            if $c = c'$ then                                   ▷ k-anonymity without modification
                $V_k \leftarrow \kappa[V_{i\cap j}]$
                break
            end if
            $\psi \leftarrow \{\}$                             ▷ Get new neighbors related to the k-group
            for all $v_j \in \kappa[V_{i\cap j}]$ do
                $V_{j\setminus i} \leftarrow G.nbrs(v_j) \setminus V_i \setminus \kappa[V_{i\cap j}] \setminus \{v_i\}$
                for all $v_l \in V_{j\setminus i}$ do
                    $\psi[v_l] \leftarrow G.nbrs(v_l) \cap \kappa[V_{i\cap j}]$
                end for
            end for
            $\eta \leftarrow \{\}$                             ▷ Filter applicable groups and neighbors
            for all $\psi[v_l]$ do
                for all $\gamma \subseteq \psi[v_l]$ if $|\gamma| = k - 1$ do
                    $\eta[\gamma] \leftarrow \eta[\gamma] \cup \{v_l\}$
                end for
            end for
            if $\exists \eta[\gamma]$ that $|\eta[\gamma]| \geq c - c'$ then
                pick $\eta[\gamma]$ where $|\eta[\gamma]| \geq c - c'$
                $V_k \leftarrow \gamma$
                $E_k \leftarrow \eta[\gamma]$
                break
            end if
        end for
        $c' = c' - 1$
    end while
    return $V_k, E_k$                                          ▷ Existing and new neighbors for k-anonymity
end procedure
```

6.4.2 Analysis of the y-identity Model

Owing to the failure of k-anonymity led to the proposal and analysis of the y-identity method. Here, the user creates y new identities and randomly assigns the privacy sensitive information to one of the identities randomly. 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. It is assumed the user is rational and optimizes for the best applying privacy-preserving settings, thus he would always choose a single identity for storing the sensitive value among all identities in all datasets; meaning that here identity separation is not limited to a single dataset.

Such an attribute can be 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). In real-life scenarios this process should be supported by an identity manager software (e.g., Scramble is such a proof-of-concept utility [42]), by which the user could be able to reveal the secret information for the selected audience with ease. An important constraint for the attribute to be hidden is that alternatives need to be credible to maintain plausibility, otherwise the attacker can easily rule out false data and learn the sensitive one. As a result, for the attacker, the social network platform and the other users separated identities would be represented as separate users with differing attributes.

Definition 5. *y-identity. A users is considered to be acting according to the y-identity model if he 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.*

In addition, instead of attacks targeting a single node (or a small group of nodes), I consider mass-attacks that aim to re-identify thousands of nodes in some sanitized networks, like attacks as Nar09 [2]. I assume that the attacker is rational, and aims for revealing 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 their separated identities in sanitized datasets (in Section 6.3 I provided an example for finding multiple separated identities by utilizing Nar09). Then, after finding these mappings for a given user, the attacker makes a decision and

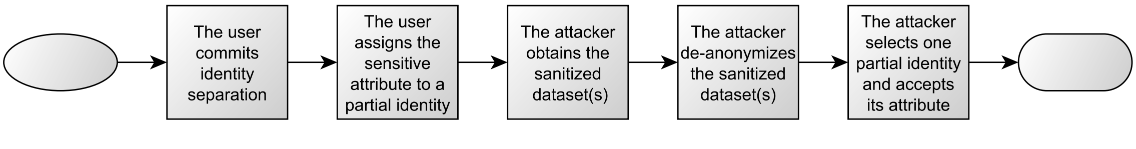

The user commits identity separation
The user assigns the sensitive attribute to a partial identity
The attacker obtains the sanitized dataset(s)
The attacker de-anonymizes the sanitized dataset(s)
The attacker selects one partial identity and accepts its attribute

Figure 6.6: Subsequent steps the participants take within the y-identity model.

either selects none, or picks one of the partial identities to be valid (i.e., learn the sensitive information). This process is illustrated in Fig. 6.6.

Formal Description of the Attack

Focusing on a given user and the attacker, we can formally describe this process similarly as a game; however, I did not always model it as a game (see the attacker model for details). Therefore, the player set $\mathcal{P}$ contains the user and the attacker.

Initially, the user v_n creates a total of y new identities (in a single or even in multiple services) denoted as $v_{n\backslash i}$, and the one having the sensitive attribute denoted $v^{\star}_{n\backslash i}$. The whole strategy set $\mathcal{S}$ can be defined as selecting one of the identities the user has, either for storing the sensitive attribute (user) or for selecting it to be valid (attacker). In some cases the attacker only has access to $\mathcal{S}' \subset \mathcal{S}$, limiting its possible decisions. It is important to note that strategic steps are taken only once. 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 learning process itself.

The user decision is modeled with $P(R = i) = r_i$, where $\sum_{\forall i} r_i = 1$ (n.b. this includes the possibility of a deterministic decision, where $\exists r_j = 1$). After these steps, in some way, the attacker obtains some of the anonymous datasets of the networks that contain these identities, and by using some background information run a structural re-identification algorithm to find all $v_{n\backslash i}$. I assume that the attacker only captures the sanitized dataset after the user committed identity separation, and knows no information about the identity separation process itself. At this point, attacker decisions are modeled with $P(Q = i) = q_i$, the probability for accepting the sensitive attribute of $v_{n\backslash i}$ to be valid. For the attacker we can allow $\sum_{\forall i} q_i \leq 1$, as some attackers might not accept any attributes to be valid at all, for instance, because all of them are in conflict the background knowledge of the attacker.

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 considered cases is provided in Table 6.1 for $y = 2$ identities within a single dataset.

Payoffs can be strongly asymmetric. For instance, a single node may not be very

		User	
		$v^{\star}_{n\backslash 1}$	$v^{\star}_{n\backslash 2}$
Attacker	$v^{\star}_{n\backslash 1}$	$u^{+}_{A}; u^{-}_{n}$	$u^{-}_{A}; u^{+}_{n}$
	$v^{\star}_{n\backslash 2}$	$u^{-}_{A}; u^{+}_{n}$	$u^{+}_{A}; u^{-}_{n}$

Table 6.1: Utility matrix ($\mathcal{U}$) for the case of $y = 2$.

important for the attacker (as being only one of hundreds of thousands), while the targeted private value can be very important for the user.

Attacker Model

In the attacker model, we can define 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 possible choices each other could make (both players know $\mathcal{S}$), a game-theoretic approach can be conveniently used for searching the best strategies.

2. *Weak attackers*, who are able to reveal some of the identities (even all of them), but are uncertain if there are any additional identities (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 $\mathcal{S}' = \mathcal{S}$. Although there are missing possible pure strategies of the user, this case could also be formalized as a game with significantly increased complexity. However, we can also model the attacker as making decisions according to a given distribution on the discovered identities. In this case, for searching the best user strategy, I 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.

For example, the distribution used for decision making by the weak attacker type can be determined by 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 algorithm works (as naive algorithms are quite deterministic in this for a given background knowledge). In case of strong attackers, I assert that they always make a choice, i.e., $\sum_{\forall i} q_i = 1$.

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 in the future. 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.

Evaluation of Strong Attackers

I model this problem 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. This *identity partitioning game* works as follows. The user assigns the sensitive information to $v_{n\setminus i}$ with probability r_i (resulting in $v^\star_{n\setminus i}$). The attacker obtains the concerned sanitized datasets, and by running a re-identification algorithm, he finds all identities, and accepts 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 [77] of this game is a pair of strategies when none of the players can increase their payoff by modify only their strategy alone. It can be easily concluded that no pure strategy equilibrium exists in this game. If the user constantly chooses the i^{th} identity as his strategy, the attacker can respond by choosing the corresponding identity, modifying the payoffs as $u_A^+; u_n^-$ favoring himself. Having any kind of response of the user, the attacker could always have a response leading to an equivalent situation.

Fortunately, John Nash have proven that in finite games a mixed strategy equilibrium should always exist [78], and here I prove the exact probabilities of the mixed equilibrium strategy.

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

$$u_A^- r_i + \sum_{\forall k \neq i} u_A^+ r_k = u_A^- r_j + \sum_{\forall l \neq j} u_A^+ r_l, \tag{6.1}$$

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 (6.1) can be simplified as:

$$u_A^- r_i + u_A^+ (1 - r_i) = u_A^- r_j + u_A^+ (1 - r_j) \tag{6.2}$$

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

$$\left(\begin{array}{ccccc|c} u_A^- - u_A^+ & u_A^+ - u_A^- & 0 & \cdots & 0 & 0 \\ u_A^- - u_A^+ & 0 & u_A^+ - u_A^- & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ u_A^- - u_A^+ & 0 & 0 & \cdots & u_A^+ - u_A^- & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{array}\right) \tag{6.3}$$

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 (6.2) can be reduced to

$$r_i = r_j. \tag{6.4}$$

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 for the attacker, which calculation will be identical 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 the intuitive approach to be the most efficient one somebody could find against strong attackers: the best strategy is to use random, equal assignment probabilities. However, as shown later, it is not necessarily also the best for other types of attackers.

Evaluation of Weak Attackers

Here, I assume that the user can assess P_i, the discovery probabilities respectively of $v_{n\backslash i}$ $(\forall i \in [1, y])$. I work with P_i in a general sense, but P_i can be derived at least of two factors: the probability that the attacker can access the dataset that includes $v_{n\backslash i}$, and additionally the probability of finding that identity. (In case of strong attackers, using these probabilities would not make sense, as the attacker is more likely to be able to calculate and use these.) In Section 6.3 I shown 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 some cases; thus in such a case, I propose to stick to the solution proposed for unknown attackers (see below).

Let us calculate the expected privacy loss. Let start with a specific case when the attacker discovers some given identities of the user v_n. 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 $(v_{n\backslash i})$ was discovered or not ($m_i \in [0, 1]$, $m_i = 1$ indicating the identity was found, and vice versa). 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 can refine the attacker decision distribution, and introduce distribution vector denoted $\mathbf{q_m}$, 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:

$$u_n^- \cdot \left(\sum_{\forall i} r_i \cdot q_i^{\mathbf{m}} \cdot m_i \right), \forall i \in [1, y] \tag{6.5}$$

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:

$$P_{\mathbf{m}} = \prod_{\forall j} ((1 - m_j) + (-1)^{(1-m_j)} \cdot P_j), \forall j \in [1, y] \tag{6.6}$$

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

$$E_w[u_n] = \sum_{\forall \mathbf{m}} \left(\left(\prod_{\forall j} ((1 - m_j) + (-1)^{(1-m_j)} \cdot P_j) \right) \cdot \left(\sum_{\forall i} r_i \cdot q_i^{\mathbf{m}} \right) \right) \cdot u_n^- \tag{6.7}$$

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

However, this formula leads to an interesting advice regarding the best user strategy.

Theorem 2. *Given a weak attacker with known $\mathbf{q_m}$ vectors (for all $\mathbf{m}$), a set of pure strategies $\mathcal{S}' \subseteq \mathcal{S}$ exists which 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 (6.7) can be rewritten in the following way:

$$E_w[u_n] = u_n^- \cdot \sum_{\forall \mathbf{m}} \left(P_{\mathbf{m}} \cdot \sum_{\forall i} r_i \cdot q_i^{\mathbf{m}} \right) = u_n^- \cdot \sum_{\forall i} \left(\underbrace{(\sum_{\forall \mathbf{m}} q_i^{\mathbf{m}} \cdot P_{\mathbf{m}})}_{\alpha_i} \cdot r_i \right) \tag{6.8}$$

Here term α_i is a known constant, thus we seek the minimum value of a linear sum with non-negative coefficients. This value is minimal when:

$$\sum_{\forall j \in \arg\min_j \alpha_j} r_j = 1, \tag{6.9}$$

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 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.

□

The conclusion of Theorem 2 is that in the case of weak attackers (w.r.t. the attacker model), in general it is advised to use pure strategies instead of mixed ones. In some specific cases, when there are multiple, equally good choices, mixed strategies can be based based on those strategies. I have provided examples on using this model for assessing strategies in Appendix A.3.

The model can be further extended by allowing 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 are in (6.7), implicitly decreasing the expected privacy loss value, too. Introducing such a state would mathematically suggest that one should maximize r_0, which is consistent with the common sense saying that if you want to have maximal privacy do not publish any sensitive content.

Most Likely Scenario: Attacker Strategy Unknown

In case of the k-anonymity setting, ideally (in Section 6.4.3, I discuss why this is not often the case), the expected privacy loss is

$$E_k[u_n] \leq \frac{u_n^-}{k}, \tag{6.10}$$

as according to the k-anonymity definition there should be at least k entities with the same quasi-identifier (including the user). The more of such entities there are, the more $E_k[u_n]$ is likely to decrease.

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, I propose to use the equilibrium strategy $r_i = \frac{1}{y}$; the following part proves why that is an appropriate choice.

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:

$$E_s[u_n] \leq \frac{u_n^-}{y} \text{ and } E_w[u_n] \leq \frac{u_n^-}{y}. \tag{6.11}$$

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

$$E_s[u_n] = \frac{u_n^-}{y}. \tag{6.12}$$

Let us check the expected privacy loss for weak attackers by substituting $r_i = \frac{1}{y}$ to (6.7):

$$E_w[u_n] = \left(\sum_{\forall \mathbf{m}} \left(\prod_{\forall j} ((1 - m_j) + (-1)^{(1-m_j)} \cdot P_j) \cdot \sum_{\forall i} q_i^{\mathbf{m}} \right) \right) \cdot \frac{u_n^-}{y} \tag{6.13}$$

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

$$E_w[u_n] \leq \frac{u_n^-}{y} \cdot \sum_{\forall \mathbf{m}} \left(\prod_{\forall j} ((1 - m_j) + (-1)^{(1-m_j)} \cdot P_j) \right) \tag{6.14}$$

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:

$$E_w[u_n] \leq \frac{u_n^-}{y}, \tag{6.15}$$

which, with (6.12), satisfies the criteria for the theorem according to (6.11). □

Theorem 3 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, as the expected privacy loss would still have a feasible higher bound.

6.4.3 Comparison of k-anonymity and y-identity

In the k-anonymity model, there were k seemingly identical users (as they are structurally equivalent), therefore choosing one can be approximated with the same probability $\frac{1}{k}$. While the expected loss of the y-identity model can be upper bounded with the expected loss for k-anonymity, for some identities the risk can be significantly higher than compared to others. Let us demonstrate this on a simple example. Let us suppose there is a user who has $y = 5$ identities, which have different discovery probabilities as $\forall i \leq 4 : P_i \ll P_5$. The user then randomly assigns the sensitive attribute to one of the identities with the same probability, $r = \frac{1}{5}$. However, while $v_{n\backslash 5}$ also has r as the other identities, getting $v^{\star}_{n\backslash 5}$ is a risky business: due to the high discovery probability it has, it is very likely that even a naive attacker could compromise the privacy of the user v_n.

These kind of risk 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; e.g., doubling the identity ($k = 2$) for $v_{n\backslash 5}$ by introducing a structurally equivalent $v_{n\backslash 6}$ but with different sensitive attribute.

Or, if possible, k-anonymity could be established by aligning partial identities to their neighborhood. The difference here to Algorithm 1 is that parameter c would be a constraint defined by the neighborhood, not chosen by the user. Fortunately, according to my measurements described in Section 6.3 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 I managed to eliminate in the 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.

6.5 Conclusion

In this section I discussed identity separation from an individual point of view: how this technique can be used to minimize personal information disclosure. I have shown that using a relatively high number of identities and deleting a low fraction of edges (best model with random deletion, $Y = 5$) provide a statistically appropriate protection in case of the state-of-the-art attack, Nar09: even regarding recall rates for privacy-defending users or the chanches the attacker has for recovering multiple identities of a user. I have also proposed a simple algorithm where decoy identities are used to bait Nar09. This technique is the first where the user has some control in choosing which identity the attacker should find.

However, the attacker can adopt his algorithm to the discussed techniques. Therefore, my goal was to provide strategies that have guarantees regarding what the adversary can discover. As possible additional improvement, I 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, I 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. I also introduced a reasonable attacker model for the problem, and proved that even if the attacker type is not known (as this is likely to happen 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. I additionally discussed that the y-identity model fixes a serious vulnerability of k-anonymity.

Finally, it should be mentioned that the y-identity model can be combined with the k-anonymity model. This can be done similarly as provided with the example in Fig. 6.4, i.e., new identities created within the y-identity model can be chosen in a way to have k-anonymity with some of their neighbors-of-neighbors. This could provide additional privacy, however, it heavily depends on the structure of the neighborhood of the node if the techniques can be combined or not.

Chapter 7

Applications and Future Work

7.1 Application of New Results

The goal of my work is to fill the gap on protecting structural information in social network based services, by providing a deep analysis of a technique that intuitively is capable of tackling such attacks: identity separation. However, adopting the proposed strategies manually is difficult, and users can not be expected to manage several partial identities on their own. 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 example, such a software could be providing anonymous access to the concerned service; e.g., including the use of anonymous web browsers [79, C9] in case of web-based social networks.

Beside, there are additional important features, such as providing a unified user interface for managing information sharing via partial identities, support for inviting users for cooperation, and having crypto protocols implemented which are necessary for operating privately. Designing such a system is possible and feasible, but a complex engineering and research task; the detailed elaboration is beyond the scope of this work. However, the design of such a system could heavily utilize results of my work; I provide some insights below how.

Results of my work showed that forcing the adoption of such an identity management tool in a top-down fashion has risks: if top nodes do not cooperate the privacy vulnerability of the whole network increases. Therefore dissemination of such a tool should be done for all participants, and *personal benefits* and *incentives* should be emphasized. I have shown that there are several suitable strategies that could work on a personal level, such as using a relatively high number of separated identities and hiding some edges. Based on results, local cooperation seems to be a suitable strategy for promotion, beneficial both for participants and the network. I have also shown that node degree and LTA can be used for assessing privacy risks, which can help in motivating users to adopt identity separation; however, in future work, these need to be normalized and transformed into objective measures.

Many people use social networks with false identities or under pseudonyms that are not related to their personal identity; while many others maintain multiple profiles for different activities. This shows a need for identity management, and I believe that there is an even greater market gap for tools that allow maintaining a valid profile and friend list available only for the desired audience, while keeping everyone else away from accessing sensitive information. Many tools can help in hiding content and profile information, but for a single exception (in [1]), there are no evaluated proposals for hiding structural meta-data. Results described in this dissertation can help within these cases, where content is hidden with tools such as Stegoweb [C5], and graph data is obscured according to the proposed strategies (e.g., best model with high number of identities). In addition, companies could also use identity separation in their software (for managing identifiers as MAC addresses) or for social network sanitization.

7.2 Future Work

In my work I found several interesting issues that should be investigated as future work. One of the core concerns is related to reversing identity separation. The simple scheme I provided in Section 6.3 could reverse identity separation in small fraction of all cases; this could be increased with further modifications. In addition, machine learning techniques could also be put to reverse identity separation, schemes such as proposed by Danezis and Sharad [37].

It would also be interesting to analyze further settings of the evaluation I provided. Within the settings I used in my work only the sanitized dataset had identity separation, while the background knowledge was a regular social network. Strategies need to be proposed to cases where the auxiliary information may also contain identity separation (n.b. as the identity separation process is assumed to be done secretly, such background knowledge could only reveal hidden attributes in the identity separated anonymous network; it would not allow linking private properties to real identities). It would also be interesting to see how the identity separation model could be extended when adding edges for deception is possible, or a larger proportion of edges can be deleted.

While my work utilized simple game theory, it could be investigated if this analysis method could be used to evaluate network protection schemes, i.e., extending the analysis to the whole network at the cost of increasing complexity significantly. In addition, the re-identification attack could also be part of the game under consideration.

Further research should also reveal how objective individual privacy assessment can be done with LTA_A and node degree.

Chapter 8

Conclusion

In this dissertation I analyzed structural re-identification attacks, and evaluated identity separation, which can be reckoned as a gradually adoptable, client side privacy-enhancing technology that could remedy the status quo.

First, I provided new details of the state-of-the-art attack. While measuring anonymity is an easy task for global identification nodes, it is not trivial for attacks capable of large-scale re-identification of networks. Therefore, I presented a way of constructing methods for measuring anonymity, and I have provided an exact measure that could be used efficiently to characterize the relative risk of re-identification of nodes within a given network. Besides I have also showed the importance of the initialization of these attacks, and experimentally analyzed how different seed selection algorithms perform compared to each other. I highlighted significant differences emerging even in the same or in structurally divergent networks.

Second, based on behavior models I proposed, I analyzed how identity separation can tackle the problem of re-identification. I presented the formula of failure probability for two seeding methods, and I analyzed using these formulas how different user strategies can be used to avoid re-identification. Results proved the major proportion of possible settings of identity separation to be effective. The method of evaluation can be used for additional global identification (or seeding) methods to evaluate the proposed privacy-protection technique in those cases, too. In the following part of my analysis, I showed that if there are enough seed nodes, without cooperation, identity separation is ineffective in limiting the attack to smaller scales. At least information leakage of nodes adopting the technique was sufficiently low. Then I evaluated multiple cooperation models, which showed that even when neighboring nodes cooperate, network privacy can be preserved more effectively, and when globally important nodes cooperate (e.g., the ones that are the most likely to be re-identified), the number of required participants becomes a fraction compared to the non-cooperative case. However, the majority of these nodes were hub nodes, and I have also showed that in all cases their participation is crucial to maintain anonymity in the network.

Third, due to the cumbersome protection of privacy of the network, I dealt with strategies that focus on the protection of the individual user. I showed multiple strategies that enable efficient information hiding from adversaries using the state-of-the-art algorithm. I also showed that using this algorithm it is hard to undo identity separation, which could only be done in a fraction of the cases, especially for users having a larger number of identities. I have also dealt with a variant of k-anonymity that could be used to have some guarantee on anonymity, however, I showed that it cannot be efficiently used against these attacks due to network diversity. In order to redress the problem, I introduced the y-identity model, and proved that there are suitable strategies enabling privacy protection with statistical guarantees.

Overall, I believe my results fill a significant gap in the research of privacy-enhancing technologies against structural re-identification attacks, as are only a very few contributions proposing (and analyzing) user centered approaches, that could be applied to existing services. These results can be used to formulate the core principles for designing privacy-enhancing systems which support identity separation in social network based services. As there are already known individual uses of identity separation (e.g., separating business and private identities), and also a considerable number of privacy-conscious users, I believe that there would be a potential demand for such a software.

In addition, these result can serve as a basis for further research as attacks evolve. While I proposed strategies that go beyond the capabilities of the state-of-the-art attack, there are certainly a variety of practical approaches that remained for future research. Personally, I think it would be important to find solutions that provide both theoretical guarantees while they are also easy to use; however, that might be harder than it first sounds.

My results also provide interesting insights for the research community. Protecting privacy is hard in general [39]: for many types of applications, it is hard to find an acceptable trade-off between the level of privacy and utility. As in my work I focused on user-centered client-side solutions, I could afford neglecting utility in the hope of achieving a sound level of privacy. Surprisingly, it turned out that even in this case tackling the state-of-the-art attack is hard in general, and only individual privacy could be protected efficiently with identity separation. These findings should encourage other researchers putting more emphasis on designing and evaluating individually applicable privacy-enhancing solutions.

Appendix A

Appendix for Further Details

A.1 Pseudo code for the undirected Nar09 algorithm

Algorithm 2: Pseudo code of the undirected Nar09 propagation phase [2].

```
Require: $\Theta$                                          ▷ Threshold for accepting new matches.
function PROPAGATE($G_{src}, G_{tar}, \mu_0$)
    $\mu \leftarrow \mu_0$
    repeat
        $(\mu, \Delta) \leftarrow$ PROPAGATESTEP($G_{src}, G_{tar}, \mu$)
    until $\Delta = 0$
end function

function PROPAGATESTEP($G_{src}, G_{tar}, \mu$)
    $\Delta \leftarrow 0$
    for all $v_{src} \in V_{src}$ do
        $S \leftarrow$ MATCHSCORES($G_{src}, G_{tar}, v_{src}, \mu$)
        if ECCENTRICITY($S$.VALUES()) $\geq \Theta$ then
            $v_c \leftarrow$ PICK($\forall v' \in S$.KEYS() : $S[v'] = max(S$.VALUES()))
            $S_r \leftarrow$ MATCHSCORES($G_{tar}, G_{src}, v_c, \mu^{-1}$)
            if ECCENTRICITY($S_r$.VALUES()) $\geq \Theta$ then
                $v_{rc} \leftarrow$ PICK($\forall v'' \in S_r$.KEYS() : $S_r[v''] = max(S_r$.VALUES()))
                if $v_{src} = v_{rc}$ then
                    $\mu[v_{src}] \leftarrow v_c$
                    $\Delta \leftarrow \Delta + 1$
                end if
            end if
        end if
        end for
        return $(\mu, \Delta)$
    end function

    function MatchScores($G_{src}, G_{tar}, v_{src}, \mu$)
        for all $v_j \in G_{tar}$ do
            $S[v_j] \leftarrow 0$
        end for
        for all $v_i \in G_{src}$.NBRS($v_{src}$)  if  $\exists\mu(v_i)$ do
            for all $v_j \in G_{tar}$.NBRS($\mu(v_i)$)  if  $\nexists\mu^{-1}(v_j)$ do
                $S[v_j] \leftarrow S[v_j] + 1.0/$SQRT($v_j$.DEGREE())
            end for
        end for
        return $S$
    end function

    function Eccentricity($S$)
                    ▷ Returns the difference of the highest and the second highest values divided by the standard deviation.
        return $(max(S) - max_2(S))/\sigma(S)$
    end function
```

A.2 Pseudo code for the Grasshopper algorithm

Algorithm 3: Pseudo code of the Grasshopper propagation phase [J2].

```
Require: $\Theta$                                              ▷ Threshold for accepting new matches.
    function Propagate($G_{src}, G_{tar}, \mu_0$)
        $\mu \leftarrow \mu_0$
        repeat
            $(\mu, \Delta) \leftarrow$ PropagateStep($G_{src}, G_{tar}, \mu$)
        until $\Delta = 0$
    end function

    function PropagateStep($G_{src}, G_{tar}, \mu$)
        $\Delta \leftarrow 0$
        $\omega_{src} \leftarrow \{\forall v_{src} \in V_{src} : v_{src} \rightarrow 1.0\}$              ▷ Initialize weights.
        $\omega_{tar} \leftarrow \{\forall v_{tar} \in V_{tar} : v_{tar} \rightarrow 1.0\}$
        for all $v_{src} \in V_{src}$  if  $\exists\mu(v_{src})$ do
            for all $v'_{src} \in G_{src}$.NBRS($v_{src}$) do
            if $\exists\mu(v'_{src}) \in G_{tar}.\text{NBRS}(\mu(v_{src}))$ then
                $\alpha \leftarrow \text{SQRT}(v_{src}.\text{DEGREE}() * \mu(v_{src}).\text{DEGREE}())$
                $\omega_{src}[v_{src}] \leftarrow \omega_{src}[v_{src}] + 1.0/\alpha$
                $\omega_{tar}[\mu(v_{src})] \leftarrow \omega_{tar}[\mu(v_{src})] + 1.0/\alpha$
            end if
        end for
    end for
    $\eta = \mu$
    for all $v_{src} \in V_{src}$ do                                  ▷ Seek new possible matches.
        $v_{tc} \leftarrow \text{BestMatch}(G_{src}, G_{tar}, \omega_{tar}, v_{src}, \mu)$
        if $v_{tc} \neq$ None then
            $v_{sc} \leftarrow \text{BestMatch}(G_{tar}, G_{src}, \omega_{src}, v_{tc}, \mu^{-1})$
            if $v_{sc} = v_{src}$ and $(\nexists\mu(v_{src})$ or $\exists\mu(v_{src}) \neq v_{tc})$ then
                $\eta[v_{src}] \leftarrow v_{tc}$
                $\Delta \leftarrow \Delta + 1$
            end if
        end if
    end for
    $\mu = \eta$
    return $(\mu, \Delta)$
end function

function $\text{BestMatch}(G_{src}, G_{tar}, \omega, v_i, \mu)$
    $S \leftarrow \{\}$
    for all $v'_i \in G_{src}.\text{NBRS}(v_i)$ if $\exists\mu(v'_i)$ do
        for all $v'_j \in G_{tar}.\text{NBRS}(\mu(v'_i))$ do
            if $v'_j \notin S.\text{KEYS}()$ then
                $S[v'_j] \leftarrow 0$
            end if
            $S[v'_j] \leftarrow S[v'_j] + \omega[v'_j]$
        end for
    end for
    if $S.\text{SIZE}() = 0$ then
        return None
    end if
    if $\text{Eccentricity}(S.\text{VALUES}()) \geq \Theta$ then
        $v_c \leftarrow \text{PICK}(\forall v \in S.\text{KEYS}() : S[v] = max(S.\text{VALUES}()))$
        return $v_c$
    end if
    return None
end function

function ECCENTRICITY($S$)
    return $(max(S) - max_2(S))/\sigma(S)$
end function
```

A.3 Examples for Calculating the Estimated Privacy Loss

In case of the y-identity model.

Table A.1: The $\mathbf{q}_m$ vectors for the first example (Section A.3.1) for all $\mathbf{m} = [m_1, m_2]$.

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 A.2: The $\mathbf{q}_m$ vectors for the second example (Section A.3.2) for all $\mathbf{m} = [m_1, m_2]$.

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

A.3.1 Example 1: Minimizing Cost in a Simple Case

Within the following examples we assume that the cost u_n^- does not differ for identities, and for keeping calculations simple we use the cost uniformly as $u_n^- = 1$ (using a negative payoff would only modify our calculation in searching for maximum points along the same principles).

Now we demonstrate the use of Eq. (6.7) in a simple example, in which there is a user with 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 A.1. By using Table A.1 and Eq. (6.7), the cost of privacy loss is characterized as:

$$E_w[u_n] = P_1 \cdot (1 - P_2) \cdot r_1 \cdot q_1 + (1 - P_1) \cdot P_2 \cdot r_2 \cdot q_2 + P_1 \cdot P_2 \cdot (r_1 \cdot q_3 + r_2 \cdot q_4) \quad \text{(A.1)}$$

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:

$$E_w[u_n] = P_1 \cdot (1-P_2) \cdot r \cdot q + (1-P_1) \cdot P_2 \cdot (1-r) \cdot (1-q) + P_1 \cdot P_2 \cdot (r \cdot q + (1-r) \cdot (1-q)) \quad \text{(A.2)}$$

This can be simplified to:

$$E_w[u_n] = \underbrace{(P_1 \cdot q - P_2 + P_2 \cdot q)}_{\text{A}} \cdot r + P_2 - P_2 \cdot q \quad \text{(A.3)}$$

Eq. (A.3) 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., $v_{n\backslash 2}$ can not be found but the attacker never chooses $v_{n\backslash 1}$. 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 his strategy for setting r if he knows (or at least have an approximation) of the parameters.

A.3.2 Example 2: Minimizing Cost Against Naive Attackers

Let us consider a user strategy against an attacker that uses a naive algorithm with a user having two identities ($y = 2$) in a single dataset. For this case we give the example $\mathbf{q}_m$ vectors in Table A.2. Here, for the sake of simplicity, we assumed that $P_1 + P_2 \leq 1$, but otherwise we could use $\frac{P_1}{P_1+P_2}$ 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:

$$E_w[u_n] = \underbrace{(P_1 - P_2) \cdot (1 + P_1 \cdot P_2)}_{\text{B}} \cdot r + P_2 - P_1 \cdot P_2 + P_1 \cdot P_2^2 \quad \text{(A.4)}$$

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 $v_{n\backslash 2}$. For $P_1 < P_2$ it should be assigned to $v_{n\backslash 1}$ ($r = 1$). Strategies proposed by this model also follows the common sense: 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 A.2 as $\mathbf{q}_{[1\ 1]} = [0.5\ 0.5]$. Here the expected cost of privacy loss is as follows:

$$E_w[u_n] = \underbrace{(P_1 - P_2)}_{\text{C}} \cdot r + P_2 - \frac{1}{2} \cdot P_1 \cdot P_2 \quad \text{(A.5)}$$

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

List of Publications

Highlighted publications are strongly related to my dissertation.

B.1 Bookchapter

[B1] K. Boda, A. M. Földes, G. G. Gulyás, and S. Imre, *Research and Development in E-Business through Service-Oriented Solutions*, ch. Tracking and Fingerprinting in E-Business: New Storageless Technologies and Countermeasures, pp. 134–166. IGI Global, 2013.

[B2] G. G. Gulyás, R. Schulcz, and S. Imre, *Digital Identity and Access Management: Technologies and Frameworks*, ch. Separating Private and Business Identities, pp. 114–132. IGI Global, 2012.

[B3] A. Kóbor, R. Schulcz, and G. G. Gulyás, *Szabad adatok, védett adatok 2.*, ch. Current threats of email - and what we can do against it (in Hungarian), pp. 315–340. INFOTA, 2008.

[B4] G. G. Gulyás, *Szabad adatok, védett adatok 2.*, ch. Using privacy-enhancing identity management in instant messaging services. (in Hungarian), pp. 285–314. INFOTA, 2008.

[B5] G. G. Gulyás, *Studies on information and knowledge processes 13., Alma Mater Series*, ch. Next generation of anonymous web browsers: a bit closer to democracy?, pp. 91–102. INFOTA, 2008.

[B6] G. G. Gulyás, *Tanulmányok az információ- ĂŠs tudásfolyamatokról 11., Alma Mater Series*, ch. Analaysis of anonymity and privacy in instant messaging services (in Hungarian), pp. 137–158. BME GTK ITM, 2006.

[B7] G. G. Gulyás, *Alma Mater sorozat az információ- ĂŠs tudásfolyamatokról 10.*, ch. Are anonymous web browsers anonymous? Analysis of solutions and services. (in Hungarian), pp. 9–30. BME GTK ITM, 2006.

B.2 Journal Papers

[J1] G. G. Gulyás and S. Imre, "Hiding information against structural re-identification," *Telecommunication Systems*, September 2014. (under review).

[J2] B. Simon, G. G. Gulyás, and S. Imre, "Analysis of grasshopper, a novel social network de-anonymization algorithm," *Periodica Polytechnica Electrical Engineering and Computer Science*, January 2015. (accepted for publication).

[J3] G. G. Gulyás and S. Imre, "Using identity separation against de-anonymization of social networks," *Transactions on Data Privacy*, January 2015. (accepted for publication).

[J4] G. G. Gulyás and S. Imre, "Analysis of identity separation against a passive clique-based de-anonymization attack," *Infocommunications Journal*, vol. 4, pp. 11–20, December 2011.

[J5] G. G. Gulyás, R. Schulcz, and S. Imre, "New generation anonymous web browsers (in hungarian)," *Híradástechnika (National Journal)*, vol. 62, no. 8, pp. 24–27, 2007.

B.3 Conference Papers

[C1] G. G. Gulyás and S. Imre, "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*, 2014.

[C2] G. G. Gulyás and S. Imre, "Hiding information in social networks from de-anonymization attacks by using identity separation," in *Communications and Multimedia Security* (B. Decker, J. Dittmann, C. Kraetzer, and C. Vielhauer, eds.), vol. 8099 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2013.

[C3] G. G. Gulyás and S. Imre, “Measuring local topological anonymity in social networks,” in *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pp. 563–570, 2012.

[C4] K. Boda, A. M. Földes, G. G. Gulyás, and S. Imre, “User tracking on the web via cross-browser fingerprinting,” in *Information Security Technology for Applications* (P. Laud, ed.), vol. 7161 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012.

[C5] T. Besenyei, A. M. Földes, G. G. Gulyás, and S. Imre, “Stegoweb: Towards the ideal private web content publishing tool,” in *Fifth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2011)* (M. Takesue and R. Falk, eds.), pp. 109–114, August 2011.

[C6] T. Paulik, A. M. Földes, and G. G. Gulyás, “Blogcrypt: Private content publishing on the web,” in *Fourth International Conference on Emerging Security Information Systems and Technologies (SECURWARE 2010)*, pp. 123–128, July 2010.

[C7] G. G. Gulyás, R. Schulcz, and S. Imre, “Modeling role-based privacy in social networking services,” in *Third International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2009*, pp. 173–178, June 2009.

[C8] G. G. Gulyás, “Design of an anonymous instant messaging service,” in *Proceedings of PET Convention 2009.1* (S. Köpsell and K. Loesing, eds.), pp. 34–40, Fakultät Informatik, TU Dresden, March 2009.

[C9] G. G. Gulyás, R. Schulcz, and S. Imre, “Comprehensive analysis of web privacy and anonymous web browsers: are next generation services based on collaborative filtering?,” in *Proceedings of the Joint SPACE and TIME Workshops 2008* (L. Capra, I. Wakeman, and M. S. Foukia, Noria, eds.), CEUR Workshop Proceedings, June 2008.

B.4 Technical Reports

[T1] T. Paulik, A. M. Földes, and G. G. Gulyás, “Publishing private data to the web (in hungarian),” tech. rep., Budapest University of Technology and Economics, 2010.

[T2] S. Dargó and G. G. Gulyás, “Using privacy-enhancing identity management in anonymous web browsers (in hungarian),” tech. rep., Budapest University of Technology and Economics, 2010.

References

[1] F. Beato, M. Conti, and B. Preneel, "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.

[2] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Security and Privacy, 2009 30th IEEE Symposium on*, pp. 173–187, 2009.

[3] "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.

[4] I. Szekely, "Building our future glass homes–an essay about influencing the future through regulation," *Computer Law & Security Review*, vol. 29, no. 5, pp. 540–553, 2013.

[5] A. Acquisti, B. V. Alsenoy, E. Balsa, B. Berendt, D. Clarke, C. Diaz, B. Gao, S. Gürses, A. Kuczerawy, J. Pierson, F. Piessens, R. Sayaf, T. Schellens, F. Stutzman, E. Vanderhoven, and R. D. Wolf, "D2.1 state of the art," tech. rep., SPION Project.

[6] S. Gurses and C. Diaz, "Two tales of privacy in online social networks," *Security & Privacy, IEEE*, vol. 11, no. 3, pp. 29–37, 2013.

[7] "diaspora*." `https://diasporafoundation.org`. Accessed: 2014-10-31.

[8] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing graphs using differentially private graph models," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, (New York, NY, USA), pp. 81–98, ACM, 2011.

[9] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility traces," in *Proceedings of the Sixteenth Annual International*

Conference on Mobile Computing and Networking, MobiCom '10, (New York, NY, USA), pp. 185–196, ACM, 2010.

[10] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: using social network as a side-channel," in *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, (New York, NY, USA), pp. 628–637, ACM, 2012.

[11] G. Danezis and C. Troncoso, "You cannot hide for long: De-anonymization of real-world dynamic behaviour," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, (New York, NY, USA), pp. 49–60, ACM, 2013.

[12] S. Ji, W. Li, J. He, M. Srivatsa, and R. Beyah, "Poster: Optimization based data de-anonymization," 2014. Poster presented at the 35th IEEE Symposium on Security and Privacy, May 18–21, San Jose, USA.

[13] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th international conference on World Wide Web*, WWW '07, (New York, NY, USA), pp. 181–190, ACM, 2007.

[14] A. Narayanan, E. Shi, and B. I. P. Rubinstein, "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.

[15] W. Peng, F. Li, X. Zou, and J. Wu, "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.

[16] P. Pedarsani, D. R. Figueiredo, and M. Grossglauser, "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, Oct 2013.

[17] S. Bartunov, A. Korshunov, S.-T. Park, W. Ryu, and H. Lee, "Joint link-attribute user identity resolution in online social networks," in *Proceedings of the sixth Workshop on Social Network Mining and Analysis*, 2012.

[18] D. Chen, B. Hu, and S. Xie, "De-anonymizing social networks," 2012.

[19] P. Jain, P. Kumaraguru, and A. Joshi, “@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, (Republic and Canton of Geneva, Switzerland), pp. 1259–1268, International World Wide Web Conferences Steering Committee, 2013.

[20] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, “Exploiting innocuous activity for correlating users across sites,” in *Proceedings of the 22Nd International Conference on World Wide Web*, WWW ’13, (Republic and Canton of Geneva, Switzerland), pp. 447–458, International World Wide Web Conferences Steering Committee, 2013.

[21] H. Pham, C. Shahabi, and Y. Liu, “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.

[22] D. Choffnes, J. Duch, D. Malmgren, R. Guimerà, F. Bustamante, and L. A. N. Amaral, “Strange bedfellows: Community identification in bittorrent,” in *Proceedings of the 9th International Conference on Peer-to-peer Systems*, IPTPS’10, (Berkeley, CA, USA), pp. 13–13, USENIX Association, 2010.

[23] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, “A practical attack to de-anonymize social network users,” in *Security and Privacy (SP), 2010 IEEE Symposium on*, pp. 223–238, May 2010.

[24] F. Reid and M. Harrigan, “An analysis of anonymity in the bitcoin system,” in *Security and Privacy in Social Networks* (Y. Altshuler, Y. Elovici, A. B. Cremers, N. Aharony, and A. Pentland, eds.), pp. 197–223, Springer New York, 2013.

[25] S. Clauß, D. Kesdogan, and T. Kölsch, “Privacy enhancing identity management: protection against re-identification and profiling,” in *Proceedings of the 2005 workshop on Digital identity management*, DIM ’05, (New York, NY, USA), pp. 84–93, ACM, 2005.

[26] R. Gross and A. Acquisti, “Information revelation and privacy in online social networks,” in *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pp. 71–80, ACM, 2005.

[27] N. Vesdapunt and H. Garcia-Molina, “Identifying users in social networks with limited information,” 2014.

[28] M. Korayem and D. J. Crandall, “De-anonymizing users across heterogeneous social computing platforms.,” in *ICWSM*, 2013.

[29] A. Zhang, X. Xie, K. C.-C. Chang, C. A. Gunter, J. Han, and X. Wang, "Privacy risk in anonymized heterogeneous information networks.," in *EDBT*, pp. 595–606, 2014.

[30] C. T. Chung, C. J. Lin, C. H. Lin, and P. J. Cheng, "Person identification between different online social networks," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, vol. 1, pp. 94–101, IEEE, 2014.

[31] L. Singh and J. Zhan, "Measuring topological anonymity in social networks," in *Granular Computing, 2007. GRC 2007. IEEE International Conference on*, pp. 770–770, IEEE, 2007.

[32] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*, vol. 8. Cambridge university press, 1994.

[33] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, (New York, NY, USA), pp. 93–106, ACM, 2008.

[34] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "hay08," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 102–114, 2008.

[35] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pp. 506–515, IEEE, 2008.

[36] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, Oct. 2002.

[37] K. Sharad and G. Danezis, "An automated social graph de-anonymization technique," in *Proceedings of the 13th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '14, (New York, NY, USA), ACM, 2014.

[38] "Nsa files: Decoded." `http://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded`. Accessed: 2014-10-31.

[39] A. Narayanan and E. W. Felten, "No silver bullet: De-identification still doesn't work," 2014.

[40] C. Aggarwal, Y. Li, and P. Yu, "On the anonymizability of graphs," *Knowledge and Information Systems*, pp. 1–18, 2014.

[41] L. A. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *Communications Magazine, IEEE*, vol. 47, no. 12, pp. 94–101, 2009.

[42] F. Beato, M. Kohlweiss, and K. Wouters, "Scramble! your social network data," in *Privacy Enhancing Technologies* (S. Fischer-Hübner and N. Hopper, eds.), vol. 6794 of *Lecture Notes in Computer Science*, pp. 211–225, Springer Berlin Heidelberg, 2011.

[43] "Stanford network analysis platform (snap)." `http://snap.stanford.edu/`. Accessed: 2014-04-22.

[44] M. Reichenbach, H. Damker, H. Federrath, and K. Rannenberg, "Individual management of personal reachability in mobile communication," in *Information Security in Research and Business*, pp. 164–174, Springer, 1997.

[45] U. Jendricke and D. Gerd tom Markotten, "Usability meets security-the identity-manager as your personal security assistant for the internet," in *Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference*, pp. 344–353, IEEE, 2000.

[46] S. Clauß and M. Köhntopp, "Identity management and its support of multilateral security," *Computer Networks*, vol. 37, no. 2, pp. 205–219, 2001.

[47] J. Hakkila and I. Kansala, "Role based privacy applied to context-aware mobile applications," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 6, pp. 5467–5472, IEEE, 2004.

[48] S. Clauß, A. Pfitzmann, M. Hansen, and E. Van Herreweghen, "Privacy-enhancing identity management," *The IPTS Report*, vol. 67, pp. 8–16, 2002.

[49] E. Franz, C. Groba, T. Springer, and M. Bergmann, "A comprehensive approach for context-dependent privacy management," in *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pp. 903–910, IEEE, 2008.

[50] M. Hansen, P. Berlich, J. Camenisch, S. Clauß, A. Pfitzmann, and M. Waidner, "Privacy-enhancing identity management," *Information Security Technical Report*, vol. 9, no. 1, pp. 35–44, 2004.

[51] "Primelife project." `http://primelife.ercim.eu`. Accessed: 2014-10-30.

[52] B. van den Berg and R. Leenes, "Audience segregation in social network sites," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pp. 1111–1116, IEEE, 2010.

[53] B. van den Berg and R. Leenes, "Keeping up appearances: Audience segregation in social network sites," in *Computers, Privacy and Data Protection: an Element of Choice* (S. Gutwirth, Y. Poullet, P. De Hert, and R. Leenes, eds.), pp. 211–231, Springer Netherlands, 2011.

[54] "Clique - a social network supporting identity management." `http://primelife.ercim.eu/images/stories/primer/clique.pdf`. Accessed: 2014-10-30.

[55] M. Kaste, "Facebook's newest challenger: Google plus."

[56] P. Adams, *Grouped: How small groups of friends are the key to influence on the social web*. New Riders, 2011.

[57] J. M. DiMicco and D. R. Millen, "Identity management: Multiple presentations of self in facebook," in *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, GROUP '07, (New York, NY, USA), pp. 383–386, ACM, 2007.

[58] S. Gurses, R. Rizk, and O. Gunther, "Privacy design in online social networks: Learning from privacy breaches and community feedback," *ICIS 2008 Proceedings*, p. 90, 2008.

[59] E. W. Weisstein, "Phase transition." `http://mathworld.wolfram.com/PhaseTransition.html`. Accessed: 2014-11-03.

[60] K. Borcea-Pfitzmann, E. Franz, and A. Pfitzmann, "Usable presentation of secure pseudonyms," in *Proceedings of the 2005 workshop on Digital identity management*, pp. 70–76, ACM, 2005.

[61] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," 2010.

[62] P. Jaccard, *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[63] G. Salton, A. Singhal, M. Mitra, and C. Buckley, "Automatic text structuring and summarization," *Information Processing & Management*, vol. 33, no. 2, pp. 193–207, 1997.

[64] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[65] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 538–543, ACM, 2002.

[66] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[67] E. Spertus, M. Sahami, and O. Buyukkokten, "Evaluating similarity measures: a large-scale study in the orkut social network," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 678–684, ACM, 2005.

[68] "Spearman's rank correlation." `http://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient`. Accessed: 2014-04-22.

[69] "The koblenz network collection." `http://konect.uni-koblenz.de/`. Accessed: 2014-04-28.

[70] L. Yartseva and M. Grossglauser, "On the performance of percolation graph matching," in *Proceedings of the first ACM conference on Online social networks*, COSN '13, (New York, NY, USA), pp. 119–130, ACM, 2013.

[71] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, pp. 452–473, 1977.

[72] L. Sweeney, "Uniqueness of simple demographics in the us population," tech. rep., Technical report, Carnegie Mellon University, 2000.

[73] C. C. Aggarwal, "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.

[74] R. Assam, M. Hassani, M. Brysch, and T. Seidl, "(k, d)-core anonymity: Structural anonymization of massive networks," in *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, SSDBM '14, (New York, NY, USA), pp. 17:1–17:12, ACM, 2014.

[75] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *In ICDE*, 2007.

[76] A. Narayanan and V. Shmatikov, “Robust de-anonymization of large sparse datasets,” in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pp. 111–125, May 2008.

[77] M. J. Osborne and A. Rubinstein, *A course in game theory.* MIT press, 1994.

[78] J. Nash, “Non-cooperative games,” *Annals of mathematics*, pp. 286–295, 1951.

[79] “Tor browser.” `https://www.torproject.org/projects/torbrowser.html.en`.

List of Figures

List of Tables

Index