ShareSafe: An Improved Version of SecGraph

Kaiyu Tang[1], Meng Han[2], Qinchen Gu[3], Anni Zhou[3*], Raheem Beyah[3*], Shouling Ji[3*]
[1]College of Computer Science, Zhejiang University
Road 38 West Lake District, Zhejiang University, Yuquan Campus, China
[tangkaiyu@zju.edu.cn]
[2]College of Computing and Software Engineering, Kennesaw State University
1100 South Marietta Pkwy, J-302, MD #9036 Marietta, GA, 30060, USA
[mhan9@kennesaw.edu]
[3]School of Electrical and Computer Engineering, Georgia Tech
Klaus 2308, USA
[sji@zju.edu.cn]
*Corresponding author: Anni Zhou, Raheem Beyah, Shouling Ji

Received October 21, 2018; revised January 22, 2019; accepted February 25, 2019; published November 30, 2019

Abstract

In this paper, we redesign, implement, and evaluate ShareSafe (Based on SecGraph), an open-source secure graph data sharing/publishing platform. Within ShareSafe, we propose De-anonymization Quantification Module and Recommendation Module. Besides, we model the attackers' background knowledge and evaluate the relation between graph data privacy and the structure of the graph. To the best of our knowledge, ShareSafe is the first platform that enables users to perform data perturbation, utility evaluation, De-A evaluation, and Privacy Quantification. Leveraging ShareSafe, we conduct a more comprehensive and advanced utility and privacy evaluation. The results demonstrate that (1) The risk of privacy leakage of anonymized graph increases with the attackers' background knowledge. (2) For a successful de-anonymization attack, the seed mapping, even relatively small, plays a much more important role than the auxiliary graph. (3) The structure of graph has a fundamental and significant effect on the utility and privacy of the graph. (4) There is no optimal anonymization/de-anonymization algorithm. For different environment, the performance of each algorithm varies from each other.

Keywords: Anonymization, de-anonymization, privacy, graph, SecGraph

1. Introduction

Nowadays, various computer networks generate lots of graph data, which could be modeled by graph structure (e.g., social networks, Internet of Things (IoT) networks, mobile traces). These graph data carries much sensitive information about users/systems. However, for research or commercial purpose, the graph data is often transferred, shared, or published to the public, research community, or commercial partners. For example, network owners often share their graph data to third parties and researchers. Such sharing behaviors bring about serious privacy leakage. To protect the privacy of users/systems, it is crucial to prevent these sensitive information from leaking during the graph transferring, sharing or publishing process.

To protect graph privacy, some anonymization techniques have been proposed to anonymize the graph, which could be categorized into six classes: Naive ID Removal, Edge Editing (EE) [1], *k*-anonymity based techniques [2]-[6], Cluster/Aggregation based techniques [7]-[9], Differential Privacy (DP) based techniques [10]-[13], and Random Walk (RW) based techniques [14]. When the graph data is anonymized, identities, name, and demographic information associated with individual node are suppressed. Such suppression is often done by removing "personally identifiable information", which will destroy the unique graph structure of the individual node. In other words, these techniques protect graph privacy by perturbing graph structure while preserving as many graph utilities as possible.

Although existing anonymization techniques can protect the graph data from a part of attacks, lots of new Structured-Basic De-anonymization attacks (DA, SDA), which the existing anonymization techniques cannot deal with, have emerged.

Based on Narayanan and Shmatikov's work [1], lots of new Structure-based De-Anonymization (SDA, when we use DA, if not specifically, it means SDA) attacks have emerged. Based on the background knowledge of attackers, these attacks could be classified into two categories: *Seed-Free* attacks, e.g., Pedarsani et al.'s attack [2], and *Seed-based* attacks, e.g., Narayanan and Shmatikov's attack [1]. Both kinds of attacks de-anonymize graph nodes via unique structural characteristic of graph nodes.

Despite lots of anonymization techniques and effective SDA attacks, considering existing anonymization techniques is not enough to defend modern SDA attacks. There still lacks a uniform and scalable platform for the graph privacy. Thus in [3], Ji et al. proposed SecGraph: a uniform and open-source Secure Graph data publishing platform. To the best of our knowledge, SecGraph is the first system that enables data owners to anonymize their graph data, measure the utility of data, and evaluate the vulnerability of data to SDA attacks.

Although SecGraph is able to anonymize graph and measure the utility and privacy of the graph data, it still has several limitations, e.g., SecGraph lacks the evaluation and quantification of the anonymous graph data for privacy security. Due to the complexity and quantity of algorithms and parameters, users have to spend much time on selecting algorithms and setting parameters for an optimal algorithm and prameters. This is not conducive for the user to deploy convenient and efficient anonymous techniques on their graph data. Besides, although SecGraph could conduct a specific security assessment via SDA attacks, it still lacks a quantitative, general, and concrete approach for security and privacy evaluations. More importantly, in SecGraph [3], Ji et al. do not consider the influence of attackers' background knowledge of seed mappings, but in reality attackers' background knowledge of seed mappings have more than 40% influence for a successful attacks. Finally, in the evaluations of SecGraph, it does not consider the influence of privacy and utility caused by the structural differences between different graphs, which is crucial and

practical. In our experiments, the structure difference between different graphs could lead to more than 30% difference for the performance of privacy and utility for anonymized graph.

Contribution. To address the above limitations in SecGraph, we design and implement the improved system of graph privacy: *ShareSafe*, which consists of five main modules: Anonymization Module (AM), Utility Module (UM), De-Anonymiza tion Module (DM), Security Quantification Module (SQM), and Recommendation Module (RM). Specifically, our main contribution is as follows.

(a) Based on SecGraph, we propose and implement ShareSafe: a safe and efficient graph data sharing platform, which consists of five modules: AM, UM, DM, SQM, and RM. ShareSafe has redesigned and improved the original modules and added the SQM and RM. We update the AM and add the latest state-of-the-art anonymization techniques to the AM. ShareSafe adds the SQM which enables users to evaluate and quantify the security of graph data. Besides, ShareSafe adds the RM to provide users a friendly and efficient way to choose optimal anonymization techniques and parameters with the aspect of utility and security requirements.

(b) We redesign the methodology of the experiments and re-evaluate the performance of utility and privacy leakage of the graph data. Besides, we evaluate the relationship among the graph structure, the utility of graph, and the graph privacy. We measure the SQM module on Facebook and Bitcoin datasets and compares the privacy leakage to realistic SDA attacks in the DM. With the SQM, We define and quantify the attacker's background knowledge and analyze the importance of the background knowledge of adversary in a successful attack.

(c) We have found that the success rate of attacks increases with the background knowledge of attackers. However, for the attacker, the help provided by the seed and the auxiliary is different from our intuitive understanding. A few seeds provide a 40% attack success rate increase far greater than the large overlapped auxiliary graph. Moreover, for different graph structures of data, the protection capabilities of various anonymization algorithms are quite different, and the leak of privacy of anonymized graph data reaches more than 30%. It shows that the modern anonymization algorithms are not specifically designed to protect certain fragile graph structures. The coarse-grained anonymization algorithm cannot meet the protection requirements of the graph data with a distinct graph structure.

The rest of this article is organized as follows. In Section 2 we introduce SecGraph and survey the most related work. In Section 3 , we introduce the ShareSafe platform. In Section 4 , we analyze the security and utility of graph data with the distinct graph structure. Finally, we conclude this paper in Section 5 .

2. SecGraph

With the large number of graph data, which may contains lots of privacy information, Many anonymization and de-anonymization algorithms have been proposed. However, there still lacks a uniform platform for the privacy of the graph data. Under this background, Ji et al. proposed SecGraph [3]: a uniform and open-source Secure Graph data sharing/publishing platforme.

2.1 System Overview

SecGraph consists of three main modules: Anonymization Module (AM), Utility Module (UM), and De-anonymization Module (DM). We briefly summarize the design and function of each module as follows.

AM: the function of AM is to anonymize graph data and generate anonymized graph data. AM contains 11 algorithms, which covers all classes of state-of-the-art anonymization techniques. Specifically, the anonymization techniques implemented are naive ID Removal, two EE based algorithms Add/Del and Switch [4], three *k*-anonymity based algorithms *k*-DA [5], *k*-iso [6], *k*-automoriphism (*k*-Auto) [7], two cluster based algorithms named bounded *t*-means clustering [8-10] and union-split clustering [10], three DP based algorithms including Sala et al.'s scheme [11], Proserpio et al.'s scheme [12, 13], and Xiao et al.'s scheme [14], and RW based algorithm [15].

UM: UM evaluates graph data's utilities between original and anonymized graphs concerning the 12 graph utilities and 7 application utilities, which, to the best of our knowledge, are enough to evaluate and characterize the usability of a graph. For more details of DM, please refer to [3]

DM: DM is used to roughly evaluate the security of graph data by using the real-world SDA attacks. The effectiveness of anonymization techniques could also be reflected via the DA evaluations. In this module, SecGraph implements 3 seed-free and 12 seed-based SDA attacks. Specifically, the implemented SDA attacks are Backstrom et al.'s attacks (BDK-the initials of the authors) [16], Narayanan-Shmatikov's attack (NS) [1], Narayanan et al.'s attack (NSR) [1], Nilizadeh et al.'s attack (NKA) [17] , Srivatsa-Hicks's three attacks (DV, RST, and RSM, respectively) [18], Pedarsani et al.'s attack (PFG) [2], Yartseva-Grossglauser's attack (YG) [19], Ji et al.'s two attacks (DeA and ADA, respectively) [20], Korula-Lattanzi's attack (KL) [21], and Ji et al.'s attack (JLSB) [22].

2.2 System Analysis

SecGraph has made a great contribution to the development of the graph privacy. However, it still inevitably has some limitations. Although SecGraph has many effective utility evaluation methods and has considered many powerful SDA attacks, it could not fundamentally evaluate/quantify the the security of graph data. Also, SecGraph contains a large number of algorithms, which is both its advantages and disadvantages. The large number of algorithms makes SecGraph more versatile and more comprehensive, but it also makes it very difficult for users to use and limits its large-scale application in practice. There are also some shortcomings in the design of the experiment for SecGraph. Nowadays most DA attacks are based on graph structure and adversary's background knowledge (especially seed mappings), which have a great impact on the success rate of SDA attacks. However, in the evaluation of SecGraph, the effect of the graph structure and seed mappings of graph data security were not considered. Modern SDA attacks are based on specific structural features, thus the relationship between the specific kind of utilities and the vulnerabilities of the graph data should also be considered. However, we do not see the relative evaluation and analysis in [3].

3. ShareSafe

To address the above limitations of SecGraph, we propose ShareSafe which includes five modules: AM, DM, UM, SQM, and RM. ShareSafe is a uniform and comprehensive graph data evaluating and sharing/publishing platform, which enables users to anonymize their graph data or evaluate the utility and security of raw/anonymized graph. We update the AM and add the state-of-the-art anonymization techniques that have emerged in recent years to the AM. Besides, unlike SecGraph, which consists of only three modules: AM, UM, and DM, we add two new modules namely *Security Quantification Module (SQM)* and

Recommendation Module (RM). For SecGraph's lack of the approach to generally evaluate and quantify the security of raw/anonymized graph data, we propose and implement SQM, which enables users to understand the potentially vulnerable graph nodes against specified graph structures. For SecGraph, a large number of algorithms and parameter settings make it flexible and comprehensive, while sometimes, it seems too complicated and inefficient. Thus we propose and implement the RM, which facilitates users who do not possess the professional knowledge. The RM enables not only quick and easy selection of anonymization techniques, but also rapid and comprehensive evaluation and quantification of the utility and privacy of graph data.

Particularly, since the need for utility evaluation of graph data is very scenario-specific and varies widely among different scenarios, it is almost impossible to add all possible utility metrics to the UM. We believe that the number of the commonly used utilities is limited, so we do not add new utilities to the UM. For modern SDA attacks, though many advanced structure-based de-anonymization attacks have been proposed in recent years, these attacks generally have little difference with the algorithms contained in the DM. The SDA attacks included in the DM are still very representative and powerful. Thus we have not added SDA attacks to DM yet.

3.1 Updates of AM

SecGraph was proposed by Ji et al. in 2015 and almost three years have passed since then. A large number of innovative graph data anonymization techniques have been proposed during this period. Therefore, the timely update to AM by adding new anonymization algorithms is very necessary. Due to time constraint, adding all of them to AM is almost impossible. At the same time, making choices among them is also very hard. Due to robustness and scalability, we choose the *k*-clique (*k*-cli) to be added to AM. *k*-clique could be classified as *k*-anonymization techniques, which protect the privacy of graph data by destroying the clique structure.

3.2 Security Quantification Module

3.2.1 Motivation

The objective of SDA attacks is to map the nodes in the anonymized graph G^a to the nodes in the auxiliary graph G^u as accurate as possible. More specifically, the framework of SDA attacks consists of two phases: *seed selection* and *mapping propagation*. In seed selection phase, attackers identify a small number of seed mappings between G^a and G^u as landmarks to bootstrap the de-anonymization. In the mapping propagation phase, attackers de-anonymize G^a through synthetically exploiting multiple graph structure similarities. Since the selection of seed mappings of most SDA attacks is almost the same, the essential difference in attacks lies in mapping propagation phase. While in mapping propagation phase, the most important and decisive role is combinations of graph structures. Therefore, our discussion of the graph privacy evaluation and quantification in SQM mainly focuses on the graph structure similarity measurements.

Because SecGraph lacks a general graph data privacy evaluation and quantification approach, we propose *Security Quantification Module (SQM)*, a module that evaluates and quantifies the security of raw/anonymized graph data and measures the effectiveness of anonymization techniques based on graph structure similarities.

3.2.2 SQM Overview

Security Quantification Module evaluates the security of raw/anonymized graphs and measures the effectiveness of anonymization techniques based on the graph structure. The SQM evaluates the security of the graph data by assessing the nodes structural similarities between G^a and G^u and quantifies the degree of the node vulnerabilities. The SQM selects the nodes with the graph structure similarity scores greater than a threshold θ in anonymized graph as the potentially vulnerable nodes. Specifically, the SQM takes as input of two graphs $G_a = (V_a, E_a)$ and $G_u = (V_u, E_u)$, structure similarity threshold θ, and the weight vector of structure similarities W. It outputs vulnerable node mappings μ between G_a and G_u. We use the ratio of the number of μ to the number of V_a as a criterion for assessing and quantifying the security/privacy of the anonymized graph data.

3.2.3 SQM Design and Implementation

The SQM is very sensitive to crucial and vulnerable graph structures of nodes in graph data. Intuitively, SQM finds node mappings using the topological structure of the graph and the information obtained from previous node mappings. SQM self-iterates for the discovery of vulnerable nodes. During the *i*-th iteration, SQM starts with accumulated node mappings μ_i between G_a and G_u. It arbitrarily chooses an unmapped node v in V_a and computes the similarity score for each unmapped node u in V_u. If the similarity score is above the threshold θ, we make u as a potential map to v. After all nodes left in V_u are considered, we add the node u_{max} with the largest structure similarity score to the vulnerable node mappings μ_{i+1}, then the next iteration starts. We describe the details as follows.

Degree Centrality and Weight Degree Centrality. The *degree centrality* is defined as the number of neighbors of the node. For instance, the degree centrality of node $v \in G_a$ is defined as $N^a(v)$ which denotes the degree of node v. Similarly, for node $u \in G_u$, the degree centrality could be denoted as $d_v = N^u(u)$. Since the graph data could be considered as a weighted graph, the weight attached on edges could provide extra information in characterizing the centrality of the node. For considering both the number of links with a node and the weight on edge, **we employ the *weighted degree centrality* in Opsahl et al. [23].** Formally, for $v \in V_a$, the weighted degree centrality is denoted as

$$wd_v = N^a(v) * \left(\frac{\sum_{v' \in N^a(v)} w_{v,v'}}{N^a(v)}\right)^{\alpha} \quad (1)$$

where $\alpha (0 \leq \alpha \leq 1)$ is a positive tuning parameter, which reflects the importance of the nodes with high degree. Larger α implies that the nodes with high degree will be considered more important.

Top-K Distance Centrality and Weighted Top-K Distance Centrality. The *top-k distance centrality* is defined as the *Cosine distance* between two top-*k* distance feature vectors. Specifically, For $v \in V_a$ (resp., V_u), its top-k distance features $Dis_k(v)$ is a *k*-dimensional vector $(dis_1^v, dis_2^v, \cdots, dis_k^v)$, where $dis_i^v (1 \leq i \leq k)$ is the distance (the shortest path) from v to the node with the $k-th$ largest degree in G_a (resp., G_u). The definition of *weighted top-k distance centrality* is slightly different with *top-k distance centrality*, which uses the weighted shortest path as the $dis_i^v (1 \leq i \leq k)$.

Closeness Centrality and Weighted Closeness Centrality. The *closeness centrality* measures how close a node is to other nodes in a graph and is defined as the ratio between the number of graphs total nodes less one and the sum of its distance to all mapped nodes. Formally, for $v \in V_a$, its closeness centrality C_v is defined as

$$C_v = \frac{|V_a| - 1}{\sum_{u \in M_a} |p^a(v,u)|} \quad (2)$$

where M_a is the mapped nodes of G_a in each iteration and $|p^a(v,u)|$ is the length of the shortest path from anonymized node v to auxiliary node u. In particular, when the graph could be modeled as weighted graph, the $|(v,u)|$ is the length of weighted shortest path from node v to node u.

Betweenness Centrality and Weighted Betweenness Centrality. The *betweenness centrality* quantifies the number of times a node acts as a bridge (intermediate node) along the shortest path between two other arbitrary nodes. Formally, for $v \in V_a$, its betweenness centrality could be denoted as

$$B_v = \frac{\sum_{i \neq v \neq j} \frac{\sigma_{ij}^a(v)}{\sigma_{ij}}}{\binom{|V_a|-1}{2}} \tag{3}$$

where $i, j \in V_a, \sigma_{ij}^a = |p^a(i,j)|$ is the number of the shortest paths between i and j in G_a and

$\sigma_{ij}^a = |\{|p^a(i,j)| \,\&\, v \; is \; an \; intermediate \; node \; on \; path \; p(i,j)\}|$ is the number of shortest paths between i and j in G_a that v lies on. For the case, when the graph could be modeled as weighted graph, the definition is lightly different for *weighted betweenness centrality*, in which the $p(i,j)$ denotes the weighted shortest path.

In summary, we use the above centrality measurements as node topological structure similarity score, which could be formally denoted as

$$s(v,u) = w_{deg} * Deg(v,u) + w_{dis} * Dis(v,u) + w_c * C(v,u) + w_b * B(v,u) \tag{4}$$

where $w_{deg}, w_{dis}, w_c, w_b \in [0,1]$ are the values indicating the weights of degree similarity, top-*k* distance similarity, closeness similarity, and betweenness similarity, respectively.

The user inputs the G_a, G_u, the similarity threshold θ, and similarity weights according to specific user scenarios. Then, the SQM calculates the structure similarity score $s(v,u)$ between node $v \in G_a$ and node $u \in G_u$, and selects the node v whose overall similarity $s(v,u)$ greater than the threshold value θ. If there is more than one node u matches the requirment, we choose the one with the highest similarity score and add v to the vulnerable nodes list. When there is no more anonymized nodes or no match score larger than θ, SQM stops the search. We consider these nodes are vulnerable in the user's scenario. Finally, we use the proportion of these vulnerable nodes in the total anonymized graph nodes as the security evaluation result of the anonymized graph.

3.3 Recommendation Module

The Recommendation Module (RM) provides users a friendly and efficient way to choose the appropriate anonymization techniques. RM helps users find the optimal anonymization techniques for their requirements of usability and security expediently. Also, RM helps users understand the distribution of utilities and the potentially vulnerable nodes in the anonymized graph. We use anonymization techniques' utility performance and security measurements of SQM as RM's evaluation criterion.

Specifically, through quantifying 12 graph utilities and seven application utilities, RM design and implement a fine-grained utility measurement metrics which enable users to flexibly and comprehensively customize scenario-specific utility measurement metrics. We use the evaluation of SQM as the security measure in RM. Thus an appropriate privacy protection level can be set according to the user's expectation. Besides, a finer granularity of privacy protection can also be achieved.

3.3.1 Architecture of RM

Fig. 1 shows the architecture of RM. Our focus is to provide a utility-preserved and multi-level privacy protected approach. The main function of the RM is to recommend the optimal anonymization techniques. If necessary, it can also perform data perturbation on users' private data with user-specified privacy and utility concern level.

RM takes two aspects of the data as inputs: raw/auxiliary graph for security evaluation and usability/privacy level controls. We discuss the details of each component of the RM as well as the interactions among them:

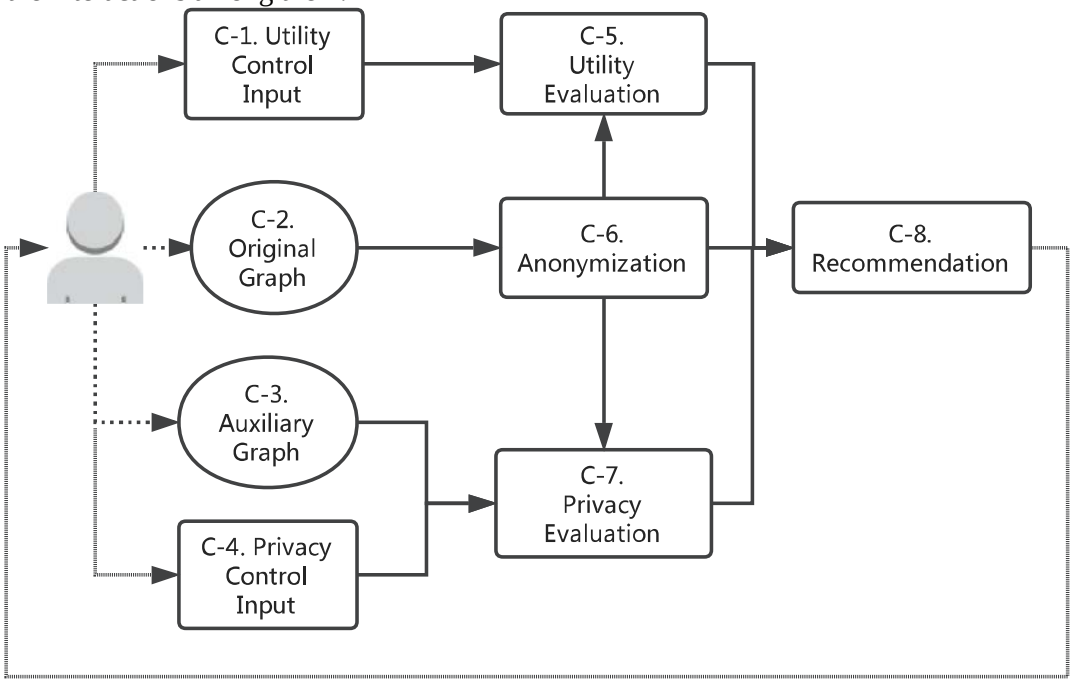

Fig. 1. Recommendation Module. The solid line represents the interactions between modules and dotted represents the interactions between user and modules

C-1. Utility Control Input *provides user an interface to obtain user's utility concern vector for user-specified utilities.* As the graph has many utilities, we allow a user to specify the utilities that he concerns and determines the importance of each specified utility.

C-2. Raw Graph *obtains user's graph data.* E.g., mobile location network, social network, and email network, etc.

C-3. Public Data *obtains public knowledge associated with the user's raw graph for security quantification.* Specifically, this component collects the auxiliary graphs that have intrinsical overlaps with the raw graph.This component also collects seed mappings between the auxiliary graphs and the raw graph. The goal of this component is to provide a user-specified security evaluation scenario, which is used to customize evaluation options for subsequent security evaluation and protection.

C-4. Security Control Input *provides user an interface to obtain a user-specified security concern level.* Specifically, C-4 provides two granularities of security control interface:

Overall (Single-level) Security Control: Provides users a single input for overall security control. The user input a decimal to represent the overall security control level, in which larger values denote the stronger security requirements.

Structure-based (Multiple-level) Security Control: Receives users' privacy control inputs as a security concern level for each graph topological structure. Users specify the

SQM security threshold θ and the weight vector for four graph structure centralities. θ represents the accuracy of privacy evaluation. When the overall privacy concern level is determined, the higher the threshold, the higher the user's security requirement.

Overall privacy provides a user with coarse-grained security control. Moreover, structure-based security control provides a user with fine-grained security control. Two different security control granularities provide a uniform and diverse protection approach for different scenarios.

C-5. Utility Evaluation *evaluates and quantifies the user-specified utilities.*This component obtains the actual utility evaluation vector by comparing the anonymized graph and the raw graph on the user-specified utilities. This component integrates 12 graph utilities and seven application utilities in UM. Also, users can freely add their own utility measurements for a more functional customization utility evaluation.

C-6. Anonymization *provides a variety of user-selectable graph anonymization schemes for the raw graph.* This component consists of 12 anonymization algorithms. The raw graph can be anonymized by a variety of selected anonymization algorithms, such as *differential privacy, k-anonymity, Clustering, Random Walk, etc.* The output of this component is anonymized graph and will be used for *C-5 Utility Quantification* and *C-7 Privacy Quantification* for further utility and security evaluation.

C-7. Security Evaluation *evaluates and quantify data privacy under the user specified security concerns.* We use SQM to measure and quantify the security of the graph.

C-8. Recommendation *selects anonymization techniques and output recommendation results, using the results of C-1 utility control input, C-4 privacy control input, C-5 utility evaluation, and C-7 privacy evaluation.* This component has two aspects:

(*1*) Analyze the degree of utility preservation and security protection based on the results of the previous components: Comparing the C-1 and C-5's results and find the anonymization techniques that satisfy the user's utility concerns. Comparing the C-4 and C-7's results and get the suitable anonymization techniques which meet the user's privacy concerns.

(*2*) Combine the utility and security performance of the graph. Then it obtains anonymization techniques that satisfy both the user's utility and security retention requirements. Also, *C-8* presents the detail performance of utility and security of appropriate anonymization techniques to users.

3.3.2 Design of RM

In this subsection, we focus on the design of recommendation module. First, we introduce several general notations. Then we present the detailed design of the main components (Privacy Evaluation (*C-7*) and Utility Evaluation (*C-5*) components) in the RM. *Notations* We define notations based on in each component:

C-1: We define a vector $\mathbf{U}$ to denote the set of utility evaluation algorithms. User's utility concern level is denoted as a vector $\mathbf{u^r} \subset \mathbf{U}$ of size *n*. The i^{th} entry $\mathbf{u^r}_i$ in $\mathbf{u^r}$ is a decimal between 0 and 1, meaning the preservation requirement for the i^{th} utility criteria in the UM. Larger $\mathbf{u^r}_i$ indicate that the more i^{th} utility is needed to be preserved in anonymized data.

C-2: User's original graph is denoted as $G = (V, E)$, which is a simple, undirected, and unlabelled graph. V is the set of vertices and E is the set of edges in G. We define $n = |V|$ to denote the number of vertices and $m = |E|$ to denote the number of edges. We use $\{i, j\}$ to define an undirected edge between vertex v_i to v_j, $deg(v_i)$ to denote the degree of vertex of v_i.

C-3: We define user collected graph as $G^{au} = (V^{au}, E^{au})$, which is a simple, undirected, and unlabelled graph. Noted that more intrinsical overlap between auxiliary and anonymized graph, more efficient for attacks and thus users will get a more powerful privacy guarantee. For seed based attacks, let $S = \{(s_1, s'_1), (s_2, s'_2), \dots, (s_k, s'_k)\}$. This priori knowledge can be used to conduct more confident ratiocination in seed based attacks. Here we only define some common and popular public information, but not public information.

C-4: Algorithm-based privacy control is denoted as a vector $\mathbf{Dr}$ of size *k*. The i^{th} entry $\mathbf{Dr}_i$ is a decimal number, representing when using the i^{th} attacks in the DM, only those nodes with similarity score greater than $\mathbf{Dr}_i$ can be considered to be vulnerable. Overall privacy control is denoted as s, which means the proportion of nodes that are successfully identified in the anonymized graph.

C-6: We define a vector $\mathbf{A}$ of size *l*. The i^{th} entry $\mathbf{A}_i$ denotes the i^{th} anonymization algorithm user chosen in the AM. Not specified, we define anonymized graph as $G' = (V', E')$. We use $G'_i = (V'_i, E'_i)$ to denote the graph anonymized by the i^{th} anonymization algorithm integrated in the AM.

Design of Privacy Evaluation

We consider using the security evaluation of SQM as our privacy notion, which not only provides a strong practical privacy guarantee but also performs better than traditional SDA attacks. Moreover, we offer both overall privacy control and fine-grained multiple level privacy control for different users and scenarios.

There are two level of privacy guarantees. (*i*) The algorithm level control denotes the confidence of node mappings in specific graph topology structure level. (*ii*) The overall level control denotes the tolerance for the proportion of nodes on all graph topology structures.

For the graph topology structure privacy control, We define the similarity function $S_i(v_i, v_j)$ to denote i^{th} graph topology structure similarity function. Let $M_{ij} = \{(m_1, m'_1), (m_2, m'_2), \dots, (m_k, m'_k)\}$ to denote the high risk of privacy node mappings found by j^{th} anonymization algorithm and i^{th} graph topology structure Thus $S \subset M_{ij}$

$$M_{ij} = \{(m_k, m'_k) | S_i(m_k, m'_k) \geq \mathbf{Dr}_i \,\&\, m_k \in |V|, m'_k \in |V'_j|\} \tag{5}$$

In overall privacy level, we define a set Ad to denote the anonymization techniques that meet user's privacy requirements. And we have that,

$$Ad = \left\{ Ad_j \mid \forall i \in [1, k],\ \frac{|M_{ij}|}{|V|} \geq s,\ Ad_j \in \mathbf{A} \right\} \tag{6}$$

Finally, we output the Ad to C-*8*.

Design of Utility Evaluation

Considering that the user's demand space is very extensive and free, we use the user-selectable utility vector $\mathbf{u^a}_i$ to measure the utility preservation of anonymized data. We define a set $Au \subset \mathbf{A}$ to denote the anonymization algorithms that meet user's requirements of utility. And we have

$$\boldsymbol{Au} = \left\{ \boldsymbol{Au_i} \mid \boldsymbol{Au_i} \in \mathbf{A} \,\&\, \mathbf{u^a}_{\boldsymbol{i}} \geq \mathbf{u}_{\mathbf{r}_{\boldsymbol{i}}} \right\} \tag{7}$$

Finally, we output the Au to C-*8*.

In C-*8*, we output all results and recommend the anonymization algorithms $\subset Au \cap Ad$: (i) anonymization algorithm and utility metric; (ii) anonymization algorithm and de-anonymization attack metric; (iii) Recommended Anonymization algorithms and their utility and privacy performance.

4. Experiment

In this section, we present our design of experiment as well as the datasets used in our evaluation. We also present and analyze our experimental results in this section.

4.1 Datasets

We use two real-world datasets that capture different graph characteristic, which is obtained from the Facebook social network [24] and Bitcoin transaction network from an Over-The-Counter(OTC) marketplace [25], which have often been used in community detection [26, 27] and graph data privacy [28, 29] research. We choose these datasets primarily because of their unique characteristics: graph from Facebook includes more users and distinct communities than Bitcoin-OTC, which, on the other hand, has only a single community (i.e., the nodes of it are all around a center) and a lower average node degree. (Unless otherwise stated, in the article, we refer to both Facebook and Bitcoin/Bitcoin-OTC as the corresponding datasets used in the experiment.)

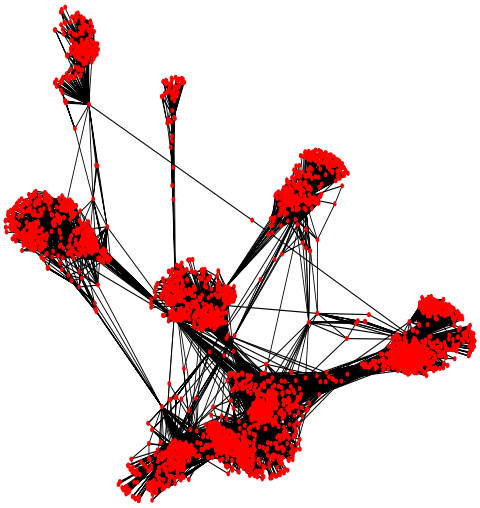

Fig. 2. Graph Structure of Facebook Social Network (node for user and edge for relationship)

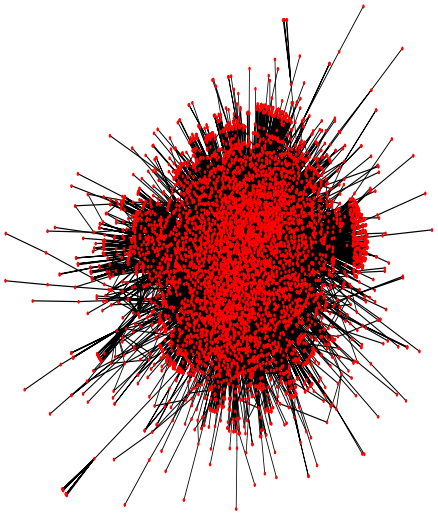

Fig. 3. Graph Structure of Bitcoin-OTC

Facebook Social Network. The facebook dataset was collected from the survey participants using a Facebook app, which consists of 4039 user nodes, 88234 edges, and 193 circles [24], and is divided into 10 ego networks. The vertices denote Facebook users, and an edge between two users represent the established friendship between them.

Bitcoin-OTC. Bitcoin is a cryptocurrency that is used to trade anonymously over the web. The dataset was created on the who-trust-whom network of people who trade using Bitcoin on a platform called Bitcoin-OTC. Bitcoin-OTC includes 5881 user nodes and 35592

edges [25]. The vertices are Bitcoin-OTC users, and an edge denotes the transactions between two users.

4.2 ShareSafe Analysis

We evaluate ShareSafe by measuring the performance of different anonymization algorithm against UM, DM, and SQM. We evaluate the utility performance of anonymization algorithm with different strength of protection. Besides, we measure the privacy exposure risk by considering different Adversary's priors in DM and SQM. We did not conduct a comparative experiment of SQM, because in addition to the SQM method, other graph privacy quantification methods are almost only theoretical methods.

4.3 Anonymization Module vs Utility Module

In this subsection, we measure each anonymization technique's performance by comparing the original and anonymized graphs' utility difference. Without loss of generality, we show 5 representative anonymization techniques that covers all types of anonymization techniques: Switch [4], *k*-clique, union-split clustering [10], improved version of Sala et al.'s DP [11-13], and RW [15]. We first anonymize original graph by anonymization algorithm in the AM. Then we measure the usability for anonymized graph by evaluating the preservation of utilities in the anonymized graph. We use the utility metric which contains 21 kinds of utility similarities between original graph and anonymized graph to measure anonymized data's usability. Specifically, when using Deg, RE, NR, RX, LCC, CC, BC, PL, JD, Infe, NC, and IM, we use cosine similarity to measure the distribution difference between original and anonymized graph. When using EV, SR, SD, and ED, we use proportion *s* to denote the difference between anonymized and the original graph. When using CD and MINS, we use Jaccard similarity.

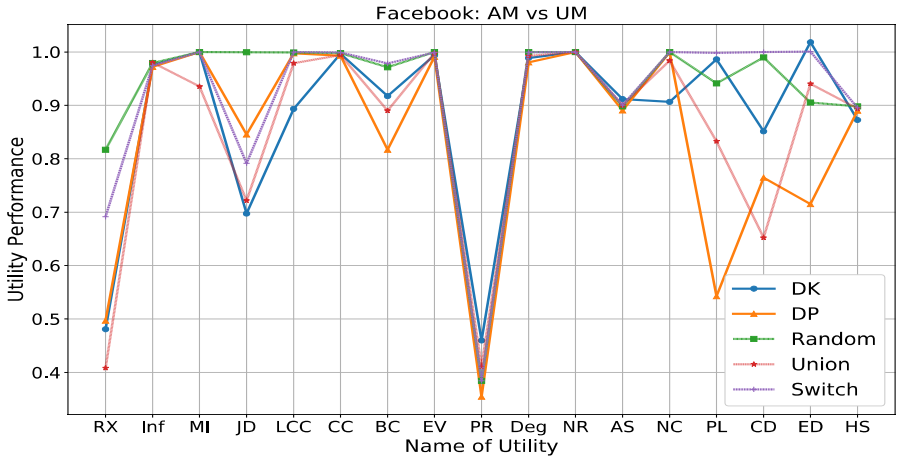

Fig. 4. Utility performance of Facebook

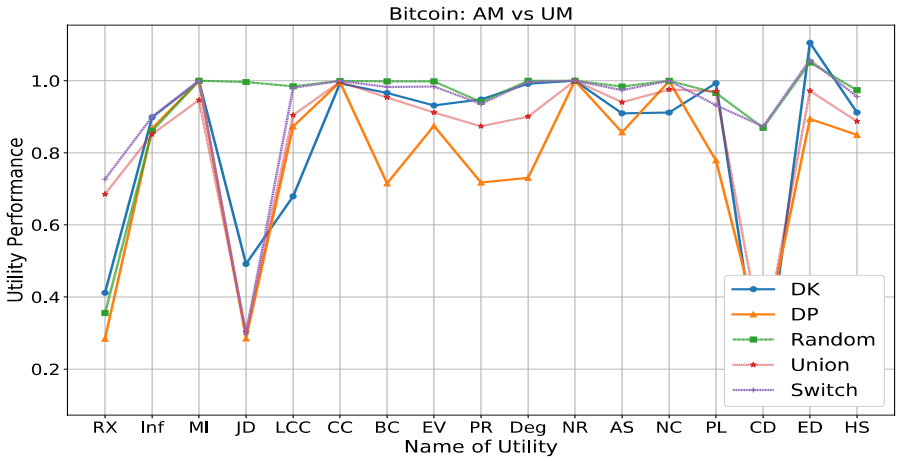

Fig. 5. Utility performance of Bitcoin-OTC

We follow the setting of original works for AM and UM evaluation and show experiment's result in **Fig. 4** and **Fig. 5**.

In general, the datasets with different structure characteristics have a very different utility performance even under the same anonymization technique. Surprisingly, the same dataset shows little difference in the performance of most graph utilities, even under different anonymization techniques. Therefore, it is necessary to choose an appropriate anonymization algorithm for different datasets and application scenarios. Our result of experiment shows that the characteristics of the dataset have a huge impact on the utility preserving of anonymized graphs.

Although the performance of different datasets has some similarities in the performance of the utility, there are still some significant differences. For example, although both Facebook and Bitcoin-OTC are underperforming on JD, it is clear that on Bitcoin, JD is less likely to be preserved, and Facebook can achieve almost twice the performance of Bitcoin. Besides, Facebook can hardly save PR under all anonymization algorithms, but PR is well preserved in Bitcoin. However, for the CD, Facebook saved much better than Bitcoin.

No anonymization scheme is optimal in preserving all graph utilities. For instance, RW performs better than every other anonymization techniques, while it perform worse than k-clique and Switch in ED.

The same anonymization algorithm performs very differently on the utility under different datasets. For instance, RW destroyed PR on Facebook, but the preservation of PR on Bitcoin can be about twice. K-clique lost almost all CD on Bitcoin but lost only about 10% of the CD on Facebook.

4.4 Anonymization Module vs De-Anonymization Module

In this subsection, we evaluate the performance of DA attacks for their basic and realistic attack capabilities. Also, we measure the privacy risk of different anonymization schedules via DM and SQM. Considering the impact of the background/prior knowledge of the adversaries on the results of attacks, we measure DA basic and realistic attack capabilities under different attacker's priors. Without loss of generality, we chose NS [1], DV [18], YG [19], ADA [20], KL [21], and JLSB [22] as attack algorithms to evaluate basic DA performance. The reason for choosing these algorithms has been described detailly in Ji et. al.

[3]. All these attacks are scalable/practical SDA attacks, and they could well represent the effects of the most advanced performance of DA attacks.

Adversarial Background Knowledge

Generic definition of the adversarial priors enables the consideration of the attacks at multiple level of privacy leakage, which is modeled by the adversary's background knowledge. We consider adversarial background knowledge from two aspects: *Auxiliary Graph* and *Seed Mapping*.

Auxiliary Graph If adversaries know a partially overlapping graph G_u with the original graph G and the real identities of the G_u, they could utilize the overlapped graph G_u to break the privacy of original graph G. We consider the $G_u \cap G = G_{com}$ and $G_{com}! = \emptyset$ and $|G_{com}| = \alpha * |G|$, where $\alpha \in [0,1]$ models the percentage of users as the Adv's background knowledge for DA attacks to G.

This prior knowledge represents that an adversary has access to the information of some users in the original graph and the corresponding subgraphs. E.g., most people will use Facebook (the social platform) and Twitter at the same time, so people's social networks on both will have some overlaps. These overlaps could be used to conduct the DA attacks on Facebook or Twitter, which may lead to serious leakage of users' privacy.

Without loss of generality, we used the methodology of the previous works [2, 3, 5, 22, 23, 25, 27]. During the generation of the auxiliary graph, we randomly sample the original graph with the probability *s*. *s* equals to the similarity α, reflecting the strength of the attacker's knowledge of the auxiliary graph.

Seed Mapping. Consider that some seed mappings are already known by attackers before the attack, in which these seed mappings could be used to iteratively de-anonymize G_a. This situation is very likely to happen in reality, because an attacker is possible to have determined part of the real mapping by being an internal employee of the network or some other means.

In practice, it is difficult to model the node mapping as the prior knowledge of adversaries, so we take a more general form. We divide the auxiliary graph G_{com} into three levels according to the node degree, and select the node mappings between G_a and G_u in each level respectively as attacker's background knowledge for seed-based DA attacks.

Experiment Methodology

We design basic DA evaluation and Advanced DA evaluation. For both basic DA evaluation and advanced DA evaluation, the attackers' background knowledge is almost the same. We randomly sample a graph with probability s from the original graph as the auxiliary graph and stratify the seed mappings from the G_{com}. The target graphs of the basic and advanced DA evaluation are different. Basic DA evaluation is used to evaluate DA performance of de-anonymization techniques. However, advanced DA evaluation is used to measure the performance of anonymization and de-anonymization techniques.

In the basic evaluation for DA attacks, the methodology we employ is generally the same as in SecGraph [3]. We use the proportion of vulnerable nodes in anonymized graph to denote the privacy loss.

For advanced DA evaluation, we use the DA attacks with various background knowledge to de-anonymize the anonymized data. First, we use six types of anonymization techniques to anonymize the data of Facebook and Bitcoin-OTC. Then, we build the adversarial knowledge by producing **Auxiliary Graph** and **Seed Mapping** for seed-based attacks. Finally, we use the SDA attacks in DM to evaluate the risk of privacy leakage of each anonymization techniques.

The auxiliary graph is obtained by randomly sampling original graph at different rate of s, and we stratify on the original graph and use the existing knowledge to get the seed mappings. Specifically, considering the strategy of seed generating which could greatly affect de-anonymization results, we employ degree stratified sampling as our seeds generating strategy, which is more practical to actual and general attack's scenarios. Finally, we employ auxiliary graph and seed mappings as attackers' priors to de-anonymize anonymized graph.

Specifically, for seed-based attacks, we feed seed based SDA attacks with 30 pre-identified seed mappings. For convenience of evaluation and comparison, we only show the highest success rate of six types of DA attacks with respect to the strength of protecting and anonymization schedule.

We show the results of basic DA evaluation in **Fig. 6** and **Fig. 7**. Besides, we show the results of advanced DA evaluation in **Fig. 8** and **Fig. 9**. Moreover, based on the results, we have the following observations.

Basic DA Evaluation

For the Facebook dataset, which has multiple centers and high degree-concentration, except for NS, the success rates of the DA attacks are slightly different. The attack success rate of each algorithm increases steadily with the sampling rate. When the sampling rate of NS is lower than 0.75 in the auxiliary graph, the attack success rate is very low. However, when the sampling rate is greater than 0.75, its success rate is much higher than other attack algorithms, which could even reach about 0.9.

For Bitcoin-OTC dataset, which is more central and with a lower average degree, the success rate of DA attacks is much lower than that in Facebook. Surprisingly, the NS which is best on Facebook performs worst on Bitcoin-OTC. Other attacks that use more complex features such as ADA, DV, JLSB, perform relatively well. When the overlap between the auxiliary graph and the original graph is greater than 0.75, the success rate of DV, ADA, and JLSB tend to stabilize at a higher level.

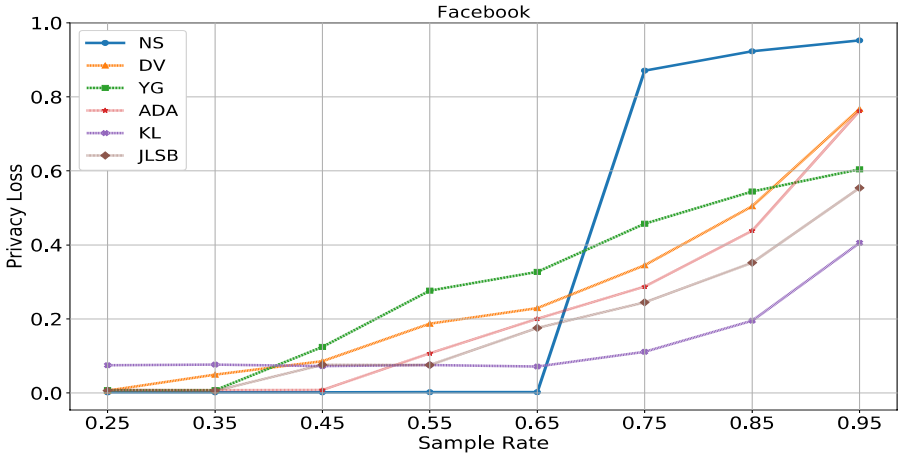

Fig. 6. Facebook Basic Privacy under the DA

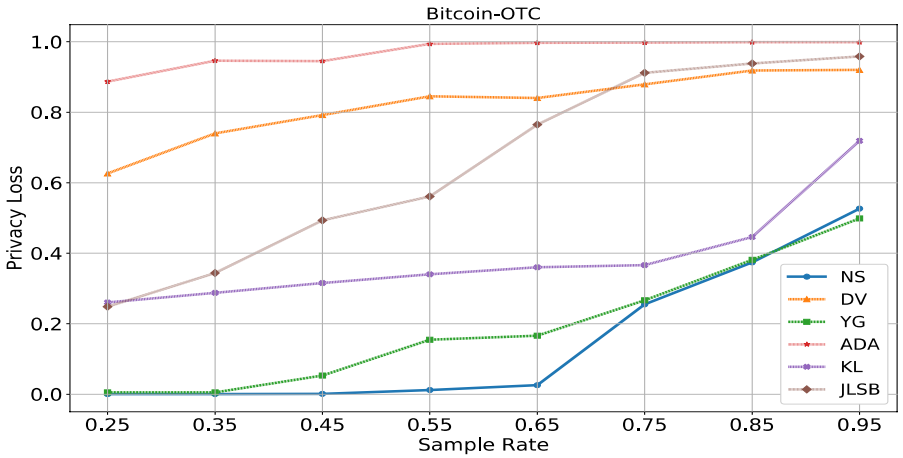

Fig. 7. Bitcoin-OTC Basic Privacy under the DA

Advanced DA Evaluation

In the advanced DA evaluation, we only show the the highest privacy loss of 5 anonymization algorithms for 6 SDA attacks with different attackers background knowledge. For the subgraph in **Fig. 8** and **Fig. 9**, each line represent the risk of privacy leakage for an anonymization algorithm.

For Facebook, we find that with the increase of the sampling rate of the auxiliary graph, the privacy loss of each anonymization schedule with the DA attack is significantly increased. Because as the sampling rate increases, the auxiliary graph and the original graph have more structural similarity, which lead to a more successful attack. TIn general, the privacy loss of the anonymized graph gradually increases with the decrease of the protection strength. Combined with the observation in basic DA evaluation, it shows that most privacy protection algorithms can protect user privacy. When the sampling rate of the auxiliary graph is greater than 0.75, the protective effects of *k*-cli, Switch, and Union are significantly less than Random Walk (RW) and DP. In most of the cases, the protection of DP is the best.

For Bitcoin, we also find that as the sampling rate of the auxiliary graph increases, the privacy loss of various anonymization algorithms also increases steadily and slowly. When the background knowledge of the attacker is fixed, the privacy loss of each anonymization algorithm is almost the same. Moreover, when the attacker's priors gradually increase, the privacy loss of each anonymization schedule begins to show a significant difference. This is particularly evident when sampling rate of the auxiliary graph reaches 0.85 and 0.95. This means that when the attacker has insufficient background knowledge, there is almost no difference in protection algorithms. However, when the attackers have more background knowledge and are powerful enough, the protection algorithms begin to gradually reveal their intrinsical deficiencies. Interestingly, the privacy loss of DP and RW are the two highest algorithms when the background knowledge is weak. But when the attacker's background knowledge is enhanced, both of them become the lowest privacy loss anonymization algorithms.

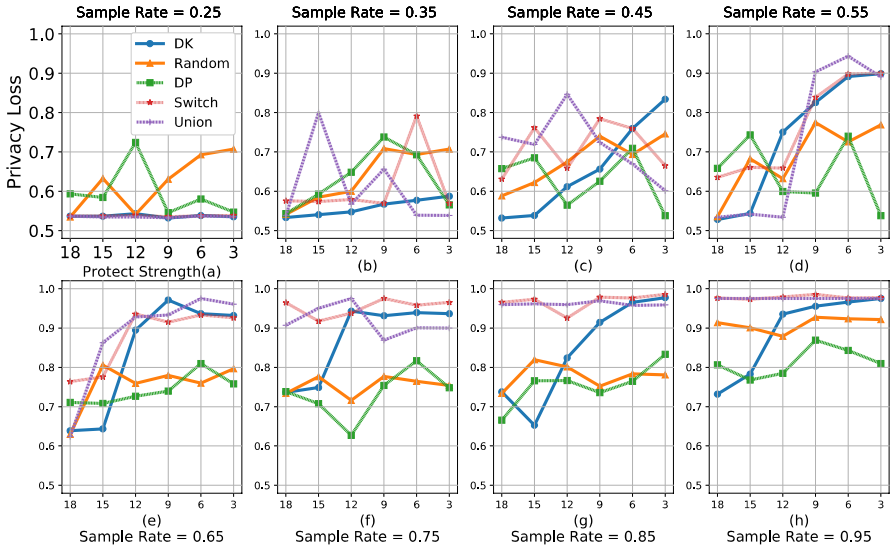

Fig. 8. Facebook Advanced Privacy under DA

In general, Facebook and Bitcoin's privacy loss has increased significantly with the increase of attackers' background knowledge. Moreover, DP and RW are the best two anonymization protection schedules in most cases (at least very close to the best privacy protection algorithm). However, the privacy loss distribution of Facebook and Bitcoin-OTC is significantly different. Facebook's overall privacy loss is much higher than Bitcoin-OTC. Moreover, the variance in privacy loss between Facebook is obviously much greater than that of Bitcoin-OTC. Besides, Facebook is more sensitive to the protection strength.

4.5 Anonymization Module vs Security Quantification Module

In this subsection, We evaluate the performance of *Security Quantification Module (SQM)* with the AM. Besides, we analyze and compare the result of DM and SQM. We demonstrate the analysis as follows.

The privacy leakage of the anonymized graph gradually increases with the background knowledge of attackers increases. When the sampling rate of the auxiliary graph is large, the difference of the privacy leakage among anonymization algorithms begins to increase. This is mainly because the attacker's attack intensity increases rapidly as the attacker's available priors increase.

With the attacker's auxiliary graph sampling rate increases, the privacy leakage of the anonymized graph also begins to increase. But in general, as the sampling rate of the auxiliary graph increases, the differentiation of privacy leakage among anonymization algorithm comes earlier than in Facebook. When the same auxiliary graph sampling rate is fixed, attacks with the seed knowledge apparently perform much better than without seed knowledge. The attacks with seed mappings have the ability to carry more powerful SDA attacks, which results in the differentiation of the privacy leakage among anonymization algorithms to occur at smaller sampling rate. Compared with **Fig. 8** and **Fig. 9**, the degree of privacy leakage of various anonymization algorithms greatly increase in seed based SDA. On average, 30 seeds, which is a very small number compared to the total number of graph vertex, are enough to increase the attacker's success rate by more than 60%. This indicates

that the seed is far more important than the auxiliary graph within the SDA attacks.

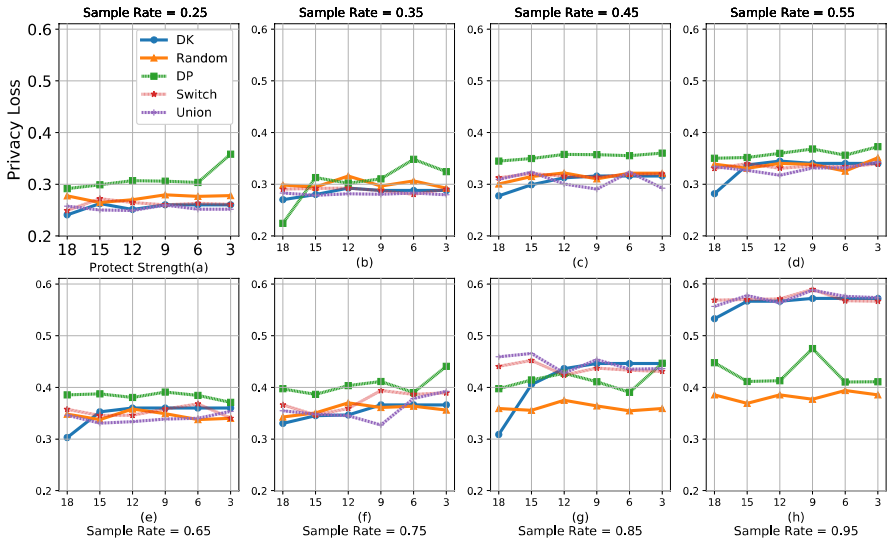

Fig. 9. Bitcoin-OTC Advanced Privacy under DA

For Bitcoin-OTC the overall privacy leakage of anonymization algorithms is small, which is around 22%. Similar to the previous observation, as the attacker's auxiliary graph sampling rate increases, the privacy leakage of the anonymized graph also slowly increases. What is different from Facebook's results is that the degree of privacy leakage of anonymization algorithms does not change significantly with the increase of auxiliary graph sampling rate. It always maintains stable in a widely differing situation. For example, Random has always been the anonymization algorithm that has the largest privacy leakage. DP has always been the smallest privacy leakage anonymization algorithm. Besides, the privacy leakage of DK, Switch, and Union have remained almost the same with different auxiliary graph sampling rates. Due to the same reason, the differentiation in anonymization algorithms is consistently large, because the attacks on Bitcoin-OTC are more likely to be successful. This is because the data of Bitcoin-OTC has a much smaller number of central nodes and the average degree is much smaller than that in Facebook.

For the seed-based SQM evaluation, the overall privacy leakage of anonymization algorithms is about 35% lower compared to Facebook (**Fig. 11**), but has increased about 40% compared with **Fig. 12**. This is consistent with the observation on Facebook. A small number of seeds provide huge advantages for SDA attack. Similar to **Fig. 12**, the privacy leakage of the anonymized graph begins to increase slowly with the attacker's auxiliary graph sampling rate increases. When the sampling rate of the auxiliary graph increases, the differentiation of privacy leakage among anonymization protection algorithms does not change significantly. Random is still the anonymization algorithm with the highest privacy leakage. DP is the anonymization algorithm with the lowest privacy leakage. Besides, the privacy leakage of DK, Switch, and Union are almost consistent under different auxiliary graph sampling rates. The gap in privacy leakage among anonymization algorithms are consistently large and stable. Bitcoin has a much smaller number of central nodes and the average degree is much smaller than Facebook, which is easier to lanch SDA attacks.

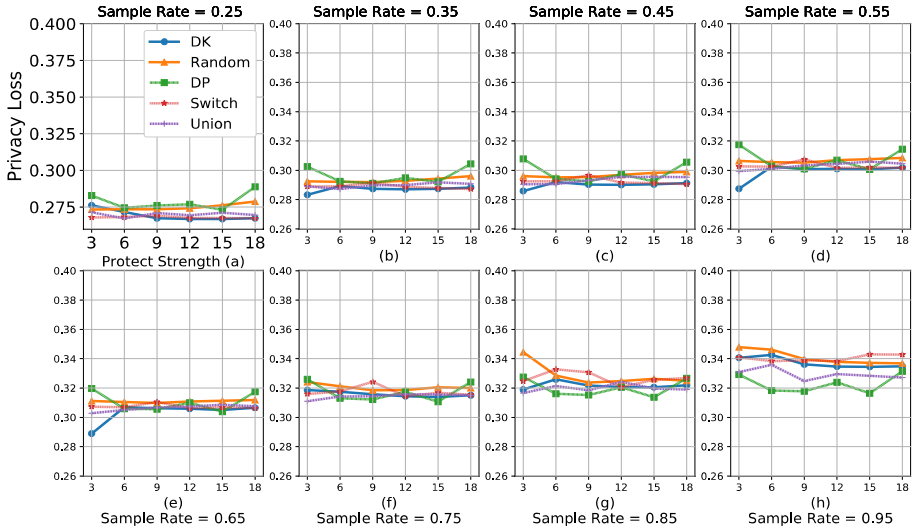

Fig. 10. Facebook Privacy without seed

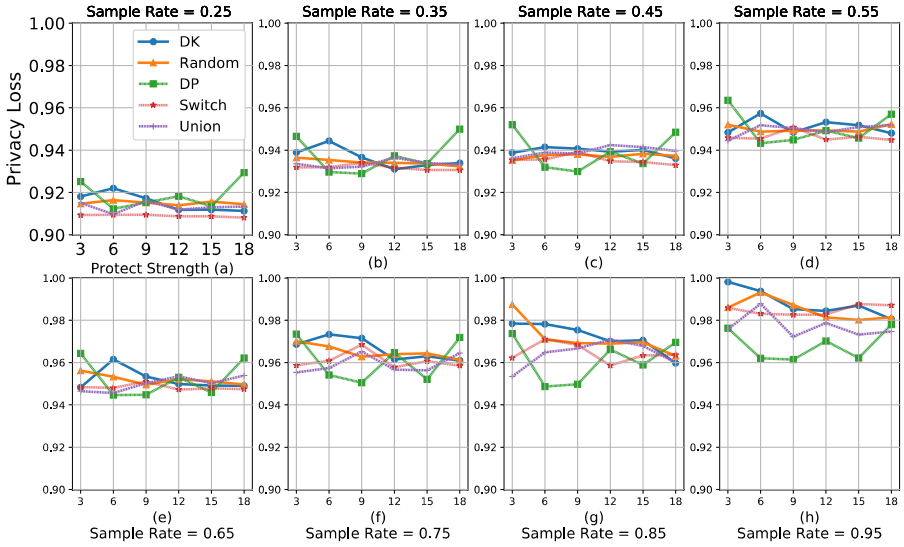

Fig. 11. Facebook Privacy with seed

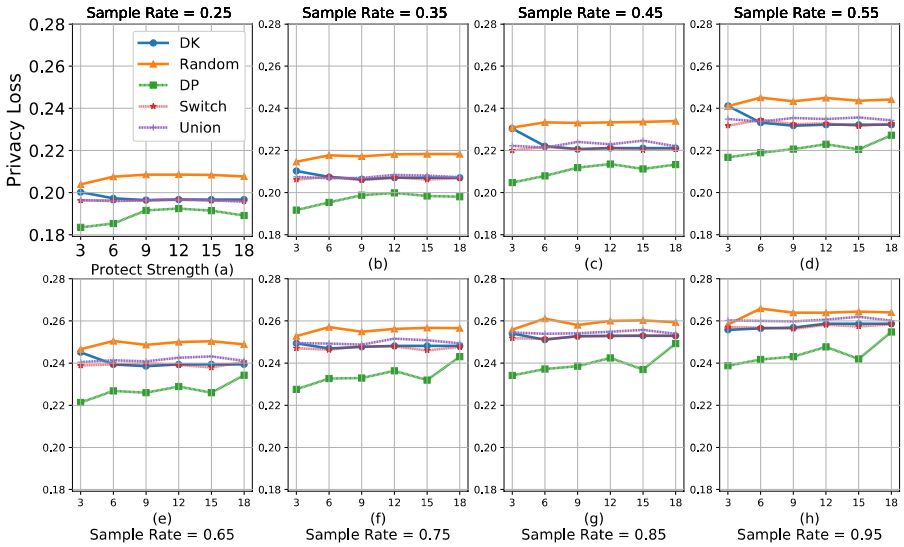

Fig. 12. Bitcoin-OTC Privacy without seed

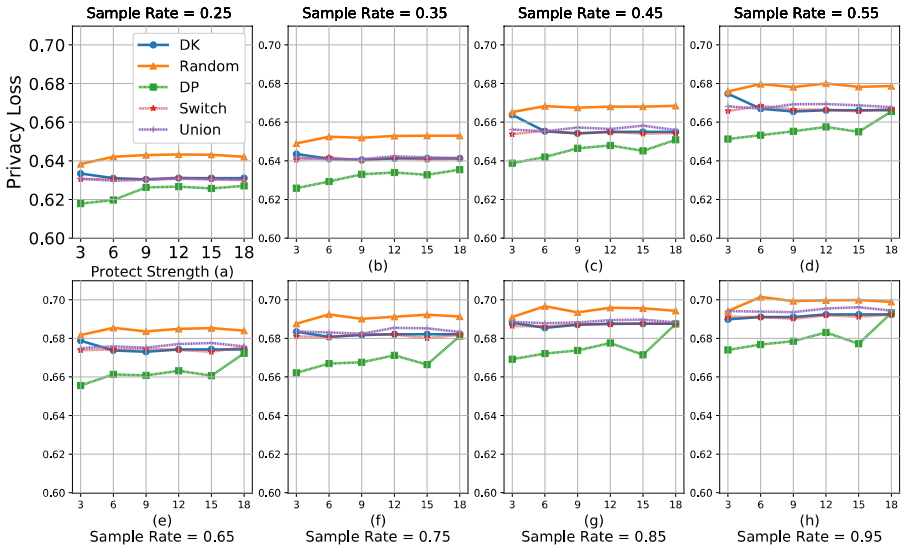

Fig. 13. Bitcoin-OTC Privacy with seed

To sum up, the privacy leakage of the anonymization algorithm increases with the attackers' background knowledge. The seed has a more important impacts than the auxiliary graph in SDA attacks. A few of seed mappings could increase the success rate about 30%. Besides, with the increase of the attacks' intensity, the difference of privacy leakage of anonymization algorithms begins to appear. However, Facebook's privacy leakage is much larger than Bitcoin-OTC. Facebook has more central nodes and a higher average degree of average. Thus Facebook is easier attacked by SDA attacks than Bitcoin-OTC.

5. Conclusion

In this paper, we redesign, implement, and evaluate ShareSafe (Based on SecGraph), an *open-source secure* graph data sharing/publishing platform. Within ShareSafe, we propose De-anonymization Quantification Module and Recommendation Module. Besides, we model the attackers' background knowledge and evaluate the relation between graph data privacy and the structure of the graph. To the best of our knowledge, ShareSafe is the first platform that enables users to perform data perturbation, utility evaluation, De-A evaluation, and *Privacy Quantification*. Leveraging ShareSafe, we conduct a more comprehensive and advanced utility and privacy evaluation. The results demonstrate that (1) As the attackers' background knowledge increases, the risk of privacy leakage of anonymized graph also increases. (2) For a successful de-anonymization attack, the seed mapping, even relatively small, plays a much more important role than the auxiliary graph. (3) The structure of graph has a fundamental and significant effect on the utility and privacy of the graph. (4) There is no optimal anonymization/de-anonymization algorithm. For different environment, the performance of each algorithm varies from each other.

Acknowledgements

The authors are very grateful to the anonymous reviewers for their time and valuable comments. The authors are also grateful to the following researchers in developing SecGraph: Shouling Ji and Stanford SNAP developers.

This work was partly supported by NSF-CAREER- CNS-0545667. Prateek Mittal was supported in part by the NSF under the grant CNS-1409415.

References

[1] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proc. of Security and Privacy, 2009 30th IEEE Symposium on*, pp. 173–187, 2009. Article (CrossRef Link)

[2] P. Pedarsani, D. R. Figueiredo, and M. Grossglauser, "A bayesian method for matching two similar graphs without seeds," in *Proc. of Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 1598–1607, 2013. Article (CrossRef Link)

[3] S. Ji, W. Li, P. Mittal, X. Hu, and R. A. Beyah, "Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization," in *Proc. of USENIX Security Symposium*, pp. 303–318, 2015.

[4] X. Ying and X. Wu, "Randomizing social networks: a spectrum preserving approach," in *Proc. of the 2008 SIAM International Conference on Data Mining*, pp. 739–750, SIAM, 2008. Article (CrossRef Link)

[5] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proc. of Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pp. 506–515, 2008. Article (CrossRef Link)

[6] J. Cheng, A. W.-c. Fu, and J. Liu, "K-isomorphism: privacy preserving network publication against structural attacks," in *Proc. of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 459–470, ACM, 2010. Article (CrossRef Link)

[7] L. Zou, L. Chen, and M. T. Özsu, "K-automorphism: A general framework for privacy preserving network publication," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 946–957, 2009. Article (CrossRef Link)

[8] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 102–114, 2008. Article (CrossRef Link)

[9] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava, "Class-based graph anonymization for social network data," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 766–777, 2009. Article (CrossRef Link)
[10] B. Thompson and D. Yao, "The union-split algorithm and cluster-based anonymization of social networks," in *Proc. of f the 4th International Symposium on Information, Computer, and Communications Security*, pp. 218–227, 2009. Article (CrossRef Link)
[11] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing graphs using differentially private graph models," in *Proc. of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pp. 81–98, 2011. Article (CrossRef Link)
[12] D. Proserpio, S. Goldberg, and F. McSherry, "A workflow for differentially-private graph synthesis," in *Proc. of the 2012 ACM workshop on Workshop on online social networks*, pp. 13–18, 2012. Article (CrossRef Link)
[13] D. Proserpio, S. Goldberg, and F. McSherry, "Calibrating data to sensitivity in private data analysis: a platform for differentially-private analysis of weighted datasets," *Proceedings of the VLDB Endowment*, vol. 7, no. 8, pp. 637–648, 2014. Article (CrossRef Link)
[14] Q. Xiao, R. Chen, and K.-L. Tan, "Differentially private network data release via structural inference," in *Proc. of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 911–920, 2014. Article (CrossRef Link)
[15] P. Mittal, C. Papamanthou, and D. Song, "Preserving link privacy in social network based systems," *arXiv preprint arXiv:1208.6189*, 2012.
[16] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," *Communications of the ACM*, vol. 54, no. 12, pp. 133–141, 2011. Article (CrossRef Link)
[17] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, "Community-enhanced de-anonymization of online social networks," in *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 537–548, ACM, 2014. Article (CrossRef Link)
[18] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proc. of the 2012 ACM conference on Computer and communications security*, pp. 628–637, 2012. Article (CrossRef Link)
[19] L. Yartseva and M. Grossglauser, "On the performance of percolation graph matching," in *Proc. of the first ACM conference on Online social networks*, pp. 119–130, 2013. Article (CrossRef Link)
[20] S. Ji, W. Li, M. Srivatsa, J. S. He, and R. Beyah, "Structure based data de-anonymization of social networks and mobility traces," in *Proc. of International Conference on Information Security*, pp. 237–254, 2014. Article (CrossRef Link)
[21] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," *Proceedings of the VLDB Endowment*, vol. 7, no. 5, pp. 377–388, 2014. Article (CrossRef Link)
[22] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural data de-anonymization: Quantification, practice, and implications," in *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1040–1053, 2014. Article (CrossRef Link)
[23] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social networks*, vol. 32, no. 3, pp. 245–251, 2010. Article (CrossRef Link)
[24] J. Leskovec and J. J. Mcauley, "discover social circles in ego networks," ACM *Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p.4, 2014. Article (CrossRef Link)
[25] S. Kumar, F. Spezzano, V. Subrahmanian, and C. Faloutsos, "Edge weight prediction in weighted signed networks," in *Proc. of Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 221–230, 2016. Article (CrossRef Link)
[26] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proc. of the sixth ACM international conference on Web search and data mining*, pp. 587–596, 2013. Article (CrossRef Link)

[27] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proc. of Data Mining (ICDM), 2013 IEEE 13th international conference on*, pp. 1151–1156, 2013. Article (CrossRef Link)
[28] N. Kökciyan and P. Yolum, "Priguard: A semantic approach to detect privacy violations in online social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2724–2737, 2016. Article (CrossRef Link)
[29] R. Trujillo-Rasua and I. G. Yero, "k-metric antidimension: A privacy measure for social graphs," *Information Sciences*, vol. 328, pp. 403–417, 2016. Article (CrossRef Link)

Kaiyu Tang got his bachelor's degree in Xi'an University of Electronic Science and Technology in 2016, and got postgraduate degree at Zhejiang University in 2019.

Meng Han currently is an assistant professor in College of Computing and Software Engineering at Kennesaw State University. He got his Ph.D. in Computer Science from Georgia State University. His research interests include Big Social Data Mining, Cyber Data Security & Privacy, and Data-driven Intelligence. He is currently an ACM member, an IEEE member, and an IEEE COMSOC member.

Qinchen Gu got his bachelor's and postgraduate degree in Electrical Engineering from Georgia Institute of Technology. His is interest in Information, network and computer security and privacy.

Anni Zhou got his bachelor's degree in Electrical and Information Engineering from Huazhong University of Science and Technology. She is now a Ph.D student in Georgia Institute of Technology.

Raheem Beyah, a native of Atlanta, Ga., serves as Georgia Tech's Vice President for Interdisciplinary Research, Executive Director of the Online Masters of Cybersecurity program (OMS Cybersecurity), and is the Motorola Foundation Professor in School of Electrical and Computer Engineering. He has held several other leadership roles including chairing ECE's Computer Systems and Software Technical Interest Group (2015 - 2017), serving as ECE's Associate Chair for Strategic Initiatives and Innovation (2016 - 2018), and serving as the Interim Steve W. Chaddick ECE School Chair during the 2018-2019 academic year. He leads the Communications Assurance and Performance Group (CAP) and is affiliated with the Institute for Information Security & Privacy (IISP). His research interests include network security, wireless networks, network traffic characterization and performance, and critical infrastructure security. He received the National Science Foundation CAREER award in 2009 and was selected for DARPA's Computer Science Study Panel in 2010. He is a member of AAAS, ASEE, a lifetime member of NSBE, a senior member of IEEE, and an ACM Distinguished Scientist.

Shouling Ji is a ZJU 100-Young Professor in the College of Computer Science and Technology at Zhejiang University and a Research Faculty in the School of Electrical and Computer Engineering at Georgia Institute of Technology (Georgia Tech). He received a Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology , a Ph.D. degree in Computer Science from Georgia State University, and B.S. (with Honors) and M.S. degrees both in Computer Science from Heilongjiang University. His current research interests include Data-driven Security and Privacy, AI Security and Big Data Analytics. He is a member of ACM, IEEE, and CCF and was the Membership Chair of the IEEE Student Branch at Georgia State University (2012-2013). He was a Research Intern at the IBM T. J. Watson Research Center. Shouling is the recipient of the 2012 Chinese Government Award for Outstanding Self-Financed Students Abroad.