IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 21, 2021, accepted March 8, 2021, date of publication March 17, 2021, date of current version October 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3066176

Trace Me If You Can: An Unlinkability Approach for Privacy-Preserving in Social Networks

KAH MENG CHONG AND AMIZAH MALIP
Faculty of Science, Institute of Mathematical Sciences, University of Malaya, Kuala Lumpur 50603, Malaysia
Corresponding author: Amizah Malip (amizah.malip@um.edu.my)

This work was supported by the University of Malaya through research under Grant GPF026B-2018.

ABSTRACT Privacy in social networks has been a vast active area of research due to the enormous increase in privacy concerns with social networking services. Social networks contain sensitive information of individuals, which could be leaked due to insecure data sharing. To enable a secure social network data publication, several privacy schemes were proposed and built upon the anonymity of users. In this paper, we incorporate unlinkability in the context of weighted network data publication, which has not been addressed in prior work. Two key privacy models are defined, namely *edge weight unlinkability* and *node unlinkability* to obviate the linking of auxiliary information to a targeted individual with high probability. Two new schemes satisfying these unlinkability notions, namely *MinSwap* and δ-*MinSwapX* are proposed to address edge weight disclosure, link disclosure and identity disclosure problems in publishing weighted network data. The edge weight is modified based on minimum value change in order to preserve the usefulness and properties of the edge weight data. In addition, edge randomization is performed to minimally modify the structural information of a user. Experimental results on real data sets show that our schemes efficiently achieve data utility preservation and privacy protection simultaneously.

INDEX TERMS Privacy, utility, social networks, unlinkability, randomization.

I. INTRODUCTION

In recent years, social networks such as Facebook, TikTok, WeChat, LinkedIn, Netflix, Google and Instagram have gained tremendous popularity as these networks support a variety of attractive features and services that help to connect the people. Rapid growth of such networks generates huge amount of sensitive individual data, which are valuable for research and development. Network data are digitally collected and the aggregated data are often published, shared or sold to third parties (such as analytics companies, marketing companies or commercial data brokers) for further analysis. Some applications of network data include analyzing the formation of communities [1], marketing and advertising [2], [3], opinion modeling [4], network information spread [5], criminal analysis [6], [7], shortest paths analysis [8]–[11] and spanning trees [12], [13]. Privacy in the applications of Ad-hoc social networks [14] and non Ad-hoc social networks [15] are also gaining the public concerns. There are laws and guidelines to restrict the types of publishable data and agreements on the usage and storage of network data, such as General Data Protection Regulation (GDPR) [16], [17] and Personal Data Protection Act [18], [19]. However, privacy breach could still occur if the data are not released under a strong privacy scheme [20].

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Hui Yeh.

A. MOTIVATION

A typical data publishing scenario involves three parties: social network users, data publisher and data recipients, as shown in Figure 1. The data publisher is a trusted entity who collects information provided by the social network users and releases the collected data to third party recipients, such as research institutes, companies and public communities. The trust relationship is not transitive to the data recipients. Some data recipients (adversaries) are not honest and attempt to infer sensitive information of a user from the published data.

Therefore, a privacy breach could occur if the personal information that a user intends to keep private, is disclosed in a published data to an entity who is not authorized to access or have the information. **In this paper, we address three privacy leaks, namely:**

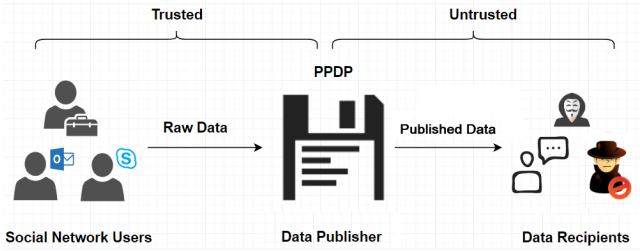

FIGURE 1. Outline of privacy-preserving data publishing (PPDP).

• **Edge weight disclosure:** Edge weight disclosure is the leak of true weightage of an edge to an adversary. For example, the communication frequency between two users.
• **Link disclosure:** Link disclosure is the inference of the true relationship between two users in a published data. For instance, a financial transaction between two users.
• **Identity disclosure (node reidentification):** Identity disclosure or node reidentification occurs when the true identity of a targeted individual is revealed by an adversary from the published data. For example, the presence of a user in a political page on Facebook.

The focus of the paper is to enable a privacy-preserved and utility-preserved weighted network data publication in an insecure environment with the assumption of an adversary attempts to attack the privacy of a user from a published data. Privacy-Preserving Data Publishing (PPDP) provides technical solutions that permit useful data mining and protect sensitive information of a user. There have been considerable interest in preserving privacy of network users associated to data publication, especially on edge weight disclosure [21]–[32], link disclosure [33]–[49] and identity disclosure problems [27], [28], [31], [32], [36]–[57]. These schemes rely upon edge weight and structural information as the background knowledge to attack the privacy of a user. The aforementioned work were mainly built to provide anonymity of the users so that the edge weight, link or identity are not identifiable within the published data. The property of unlinkability proposed in our work offers a stronger layer of privacy. It implies anonymity and further requires that the relation between the sensitive information of a user and the background knowledge are indistinguishable to an adversary. **Unlinkability provides higher privacy protection but has not been considered in the context of weighted network data publication.**

B. CONTRIBUTIONS

This paper aspires to incorporate unlinkability in weighted network data publication for a secure and useful sharing of network data. Specifically, two new privacy models are defined to address the unlinkability component in weighted network data publication, namely ***edge weight unlinkability*** and ***node unlinkability*** Furthermore, two new privacy schemes are designed based on the proposed models to satisfy different privacy and utility goals of the data publication.

The first scheme called ***MinSwap*** is proposed based on *edge weight unlinkability* to address edge weight disclosure by breaking the association between the weights and its values. The edge weight data are modified based on the idea of data swapping to fully preserve its statistical properties, including the distribution, mean, standard deviation and other statistics.

We propose another new scheme called ***δ-MinSwapX*** based on *node unlinkability* to address edge weight disclosure, link disclosure and identity disclosure simultaneously. The edge weight data are perturbed to other near values from the same data set to preserve the shortest path length. Randomization which includes selective edge deletion and random edge and node addition are deployed to prevent identity disclosure and link disclosure that rely upon edge weight and structural data as the adversary's background knowledge. Selective edge deletion allows the data publisher to minimize the distortion on network structure as the important edges can be well-preserved in the published data. The randomness is inserted during the edge and node addition phases to increase the uncertainty of an adversary in reidentifying the true identity and link, regardless of the background knowledge an adversary may possess. This efficiently protects a user against privacy leaks as auxiliary structural data provide little useful information about the true nodes in the published data.

In summary, we make the following contributions:

1) We define ***edge weight unlinkability*** and design a greedy algorithm, namely ***MinSwap*** to generate anonymized data that resist edge weight disclosure.
2) We define ***node unlinkability*** and design a greedy algorithm, namely ***δ-MinSwapX*** to generate anonymized data that resist identity disclosure, link disclosure and edge weight disclosure simultaneously.
3) We deploy data swapping, perturbation and randomization to minimally modify original network data to enhance the data utility preservation.
4) We provide a thorough analysis on the anonymization strength of the proposed work and present extensive experiment results on scalable real data sets to validate the efficiency of our schemes.

The rest of this paper is organized as follows. Section II discusses the research scope of our work. Section III gives a brief review of related work associated to privacy-preserving edge weight anonymization and structural anonymization schemes in social networks. Section IV defines two new privacy models, namely *edge weight unlinkability* and *node unlinkability*. Section V and VI elaborate on the proposed schemes for anonymizing network data. Section VII presents an extensive evaluation of the proposed algorithms using scalable real data sets in terms of security, efficiency and utility. Finally, section VIII concludes the paper.

II. RESEARCH SCOPE

In this section, we discuss the problem setting of a weighted network data publication. We present a non-directed and

weighted network model. We also define the capability of an adversary and how the adversary would utilize the auxiliary background knowledge to attack the privacy of users. In addition, we elaborate upon the desired privacy and utility objectives of the data publication.

A. NON-DIRECTED AND WEIGHTED SOCIAL NETWORK

We present a non-directed and weighted graph $G = (V, E, W)$ using Figure 2 as an illustrative example. The nodes of the graph, $V = \{v_1, v_2, v_3, \ldots, v_n\}$ denote meaningful entities from the real world such as individuals, organizations and communities. An edge $e_{i,j} \in E$ is an association between two nodes $v_i \times v_j \in V \times V$ such as friendship, partnership, co-authorship, co-workership and transaction between any two entities.

A non-directed graph consists of edges that do not have a direction (for instance, a mutual friendship). In a weighted network, each edge $e_{i,j}$ is associated with a weight $w_{i,j} \in W$ which represents the strength of connection between nodes v_i and v_j, such as the communication frequency between individuals, degree of friendship, trustworthiness and transaction amount.

B. ADVERSARY's BACKGROUND KNOWLEDGE

An adversary requires some background knowledge to attack the privacy of a target user in the published network. **In this paper, we assume that an adversary may possess partial or complete edge weight and structural information of some real-world target individuals.**

1) **Edge weight information**. Value or weightage attached to the edge, which represents the intensity and strength of the connection.
2) **Structural information**. The information about the neighbours of the target node and how these neighbours are being connected, which includes:
   a) Degree of node A, D_A: The number of edges connected to node A.
   b) Degree pair of A and B, (D_A, D_B): The degree information of node A and B.
   c) Degree sequence, **d**: A monotonic non-increasing sequence of the degree of all nodes in the network.
   d) 1-neighbourhood graph, G_A: The structural graph of node A up to the first neighbourhood of A.
   e) Subgraph of node A, S_A: A partial network graph that involves node A.

We focus on these two types of background knowledge as commonly deployed in the current literature [23], [26], [27], [32], [37]–[39], [50]–[57]. It is relatively less difficult to collect accurate edge weight information and structural graph of a targeted individual [32], [55], compared to other types of implicit information (such as eigenvector, betweenness and closeness centrality).

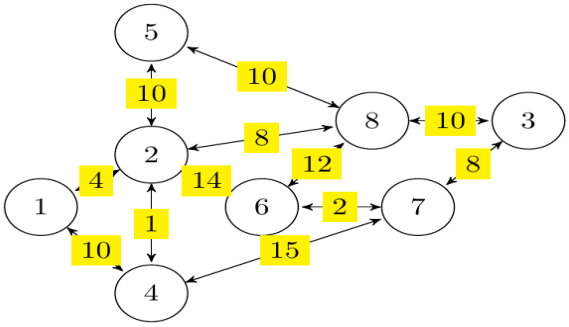

FIGURE 2. An example of a weighted and non-directed social network.

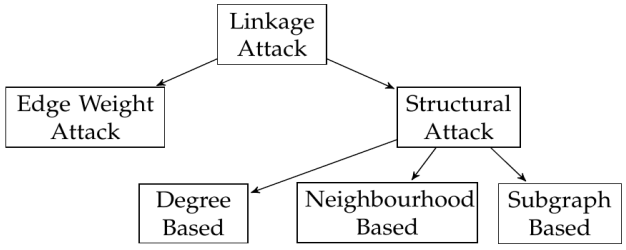

FIGURE 3. Privacy attack models.

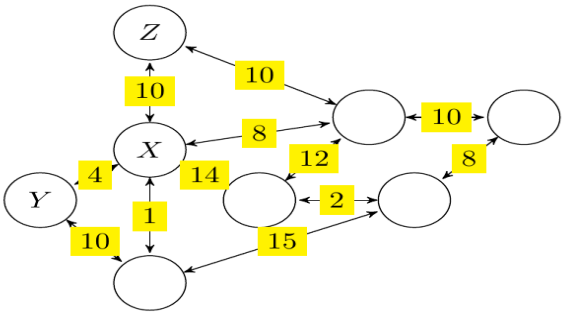

FIGURE 4. Naive anonymized weighted social network of Figure 2.

C. LINKAGE ATTACK

Linkage attack is one of the major privacy attack models in network data publication [15], where an adversary attempts to match the auxiliary background knowledge obtained from external resources to the published data in order to learn some useful information about a target victim. In Figure 3, linkage attack could be categorized as edge weight attack and structural attack according to the types of background knowledge summarized in section II.B. As the published data consists of edge weight and structural data only, other auxiliary information (such as the node label and edge label) provides very little additional information about the nodes in the published data.

Figure 4 shows a naively anonymized network of Figure 2, where the identities of all nodes are hidden. However, it is insecure when an adversary learns that node X has two connections of edge weights 1 and 4, then X's true identity (node 2 in Figure 2) is revealed. In some cases, edge weight and structural information are combined to reidentify the target. For instance, although node Y and Z in Figure 4 have

similar degree, 1-neighbourhood graph and subgraphs (and thus invulnerable to respective structural attacks), these nodes can be distinguished if an adversary possesses additional background knowledge of edge weight data.

D. PRIVACY AND UTILITY GOALS

We consider the data publishing problem where a publisher attempts to release a secure anonymized version of G, denoted by G', to serve a variety of data analysis.

The published data is said to be **privacy-preserved** if an adversary cannot infer the identity, link and edge weight values of a network user from the released data with high probability. The user's privacy is protected by limiting the ability of an adversary to infer this information, given that the adversary has full access to the published data G' and some available background knowledge.

Given an arbitrary query to an original database and its anonymized database, the outputs of query to both databases should be almost similar, that is, the difference between the outputs should be less than a parameter. A **utility-preserved** anonymized data could be produced by minimally modifying the edge weight data and network structure so that the published data remain accurate and meaningful in the data mining process. In this paper, we assume that the published data are utilized for several analyses, which include statistical analysis, shortest path length analysis and network centrality analysis [21], [23], [24], [29], [32], [42].

III. RELATED WORK

In this section, we present a comprehensive literature review on the topics related to PPDP in social networks. Particularly, we focus and discuss relevant structural and edge weight anonymization schemes that address identity disclosure, link disclosure and edge weight disclosure in social networks.

A. STRUCTURAL ANONYMIZATION

Structural anonymization schemes modify the structure of a network to prevent identity disclosure that is based on structural information as adversary's background knowledge and to address link disclosure. The schemes can be grouped under three main classifications: graph modification, clustering based method and differential privacy.

1) GRAPH MODIFICATION

Graph modification anonymizes a network by adding, deleting or switching edges or nodes in the original graph. Although similar techniques were deployed in the literature, these techniques are the basic tools used to generate different publishable graphs that satisfy different privacy and utility requirements. Graph modification can be further classified as randomization, which performs graph modification randomly and k-anonymization method, which performs graph modification to meet some desired constraints.

i) Randomization: There are different randomization approaches proposed to protect the identity and link privacy of a user [33]–[35], [37], [39]. In [33], a randomization scheme was proposed to preserve the spectrum of a graph (the set of eigenvalues of the graph's adjacency matrix) which is important to some topological properties of the graph. [34] focused on the link privacy protection and presented a neighbourhood randomization scheme which randomizes an edge by restricting the randomization to the neighbouring nodes. Hence, the network structure could be preserved to a greater extent when the structural proximity of nodes is considered. A k-candidate anonymity [37] was proposed to tackle the node reidentification attack such that there exist at least k different nodes that match every structural query over the graph. [39] proposed Bernoulli distribution to modify the edges instead of random edge addition and deletion. Bernoulli trial is deployed to determine which edge should be added or removed from the network.

Randomization does not focus on the adversary's background knowledge as the sensitive information of a user in the randomized graph are protected through the random process that modified the graph. Thus, an adversary cannot utilize the structural information to reidentify an individual from the published data as the association rules between the background knowledge and the sensitive information are dimmed. Furthermore, the presence of link cannot be inferred with high probability as randomness is deployed in the published data. The confidence level in inferring the identity, link and sensitive information of a user is bounded by a privacy level, which is affected by the amount of randomization. The data utility after randomization can only be evaluated empirically.

ii) k-anonymization Method: In k-anonymization method, the proposed schemes modify the edges and nodes in the network to produce multiple indistinguishable nodes and edges with respect to certain privacy requirements. Different assumption of adversary's background knowledge leads to different expectation of privacy criteria.

a) Degree Based Anonymization: A graph-anonymity model called k-degree anonymity was proposed in [52] to guarantee that there are at least k nodes with the same degree in the published graph. Meanwhile, a k^2-degree anonymity [53] requires that for every node with an incident edge of degree pair (D_A, D_B), there exist at least k-1 other nodes with the same degree pair in the published network. Degree of a node provides a limited structural information of a target victim. An adversary with such background knowledge is weak as the degree information can be modified easily by adding or deleting nodes and edges from the original graph. Although the schemes are invulnerable to degree attack, they are insecure against other stronger structural attacks.

b) Neighbourhood Based Anonymization: A k-neighbour- hood anonymity model [54] was proposed to guarantee that there exist at least k indistinguishable nodes in the published graph, such that the 1-neighbourhood graphs of each of the k nodes are all similar. Moreover, [55] combined both conventional k-anonymity [58] and ℓ-diversity [59] in anonymizing the social network data, such that the published

graph satisfies k-neighbourhood anonymity and contains at least ℓ different node labels. Hence, it renders stronger privacy level to the users.

c) Complete Structural Based Anonymization: In [56], a k-automorphism was proposed to defend against reidentification attacks using the structural information of node, which include node's degree, 1-neighbourhood graph, subgraph and hub fingerprint. Hub fingerprint is the distance between a hub (a node with high degree exceeding the average degree of the network) and other nodes. The k-automorphism is a strong privacy model as it guarantees that there are at least k indistinguishable nodes in the network in terms of their structural information. Hence, an adversary cannot reidentify any individual with a confidence level of higher than $1/k$ using the structural information as background knowledge. In [38], a k-isomorphism was proposed to enhance the ability of k-automorphism in link protection. The scheme creates k isomorphic subgraphs through edge additions. Two graphs are said to be isomorphic if the graphs contain the same number of nodes and the nodes are connected in the same pattern.

k-anonymization method incurs unnecessary information loss when the privacy parameter k is high. More edge modifications are performed to achieve k indistinguishable nodes. This would significantly compromise the network properties as well as the data usefulness. If the privacy parameter is low, the schemes would provide insufficient privacy protection to the users. An optimized parameter is required to provide sufficient privacy and utility level. However, the computation of the optimized privacy parameter is shown to be NP-hard in k-anonymization [60]. Therefore, modification of k indistinguishable nodes with respect to the structural graph is practically infeasible due to the high cost and high computational complexity of finding an optimal solution to the algorithms, especially when the network is scalable.

2) CLUSTERING BASED METHOD

Clustering based method involves the process of clustering nodes and edges into groups that are called supernodes and superedges, subject to some constraints on the characteristics of the nodes and edges [31], [45], [51], [57]. This approach achieves high privacy level. However, it provides a low utility as the data are changed extensively and becomes useless for certain studies. The graph is shrunk post-anonymization and most of the local structures are difficult to be analyzed.

3) DIFFERENTIAL PRIVACY

Differential privacy [61] provides a formal privacy guarantee to the nodes of a database, regardless of the auxiliary information available to an adversary. It guarantees that an adversary in possession of the released results is not able to determine the existence of an individual in the original database. Therefore, the released results provide meaningful interpretations about the underlying population statistics of the database but obscure the presence of any individual. The notion of differential privacy was adapted to network data and several new privacy definitions were formalized.

In edge differential privacy [40], two graphs G and G' are said to be edge neighbours if G' can be obtained from G by deleting or adding k arbitrary edges from G. Hence, edge differential privacy guarantees that an adversary is not able to infer the existence of a particular edge in an original database G with high probability. A local differential privacy model was proposed to preserve community structure information of a centralized and decentralized social graph with higher accuracy [41], [42].

In node differential privacy [40], two graphs G and G' are said to be node neighbours if G' can be obtained from G by deleting or adding a single node including all its adjacent edges from G. Hence, node differential privacy assures that an adversary is not able to infer the existence of a target node in an original database G with high probability. Research on node differential privacy mainly focused on improving the accuracy of publishing the degree distribution of a graph [43], [44].

A degree-differential privacy graph generation model with field theory was presented to preserve the true edges of a graph [46]. Differential privacy was deployed to add Laplace noise to the nodes' degree. The edges are then reconstructed using the proposed field theory model. A fake edge between existing nodes is generated with high probability when the interaction force between the nodes is relatively large. Hence, the impact on the structure of the graph is reduced.

Meanwhile, a random matrix approach that achieves differential privacy was proposed to publish eigenvector of a graph [47]. Two Gaussian random matrices are added to the adjacency matrix of a graph to introduce a small amount of random projection and random perturbation. Then, the projected and perturbed matrices are released as published data.

A differential privacy scheme based on graph abstraction models was proposed [48], which utilizes the dK-1, dK-2 and dK-3 series. The dK-1 level represents the degree distribution, the dK-2 level is the joint degree distribution and the dK-3 level contains the number of wedges and triangles. A differentially private noise is added to the dK-2 level of an original graph to obtain a perturbed dK-2 level, which is then used to compute the corresponding new dK-1 and dK-3 levels. Hence, a new graph is generated by combining the structural information of the three dK levels.

Differential privacy, randomization and clustering were combined to propose a PBCN (Privacy Preserving Approach Based on Clustering and Noise) [49]. The nodes are clustered into groups based on the similarity of the degree, followed by addition of Laplace noises to the degree sequence of each group. A new graph is reconstructed using the perturbed degree sequences. However, the true nodes with low degree are likely to be deleted and a number of fakes nodes are injected into the graph for fake edge addition.

Differential privacy is a strong model as it does not depend on the background knowledge of an adversary. However, the

main drawbacks of differential privacy model are presented on the utility aspect. Randomization and k-anonymization methods release a privacy-preserved graph which can be studied in place of the original database, to allow a broader range of analysis. Nevertheless, the released results under a particular differential privacy model can only serve a specific query. Furthermore, differential privacy is highly inaccurate to queries with high sensitivity. The sensitivity of a query is the largest possible difference that one data point can effect on the result of that query, for any data set. Instances of high sensitivity queries include the computation of clustering coefficient, path length distribution, betweenness distribution and closeness distribution.

4) OVERALL DISCUSSION ON STRUCTURAL ANONYMIZATION

The privacy protection is guaranteed in the structural anonymization schemes above. However, important nodes and edges are not guaranteed to be preserved in the published data. **Our work fills the gap by proposing a new randomization technique that incurs a lower utility loss.** This is achieved by considering edge deletion based on the importance of edges in the original network, such that essential edges are preserved in the published data. Hence, this may preserve network centrality to a greater extent.

B. EDGE WEIGHT ANONYMIZATION

The edge weight anonymization schemes modify the edge weight data to prevent edge weight disclosure and identity disclosure, which can be categorized under three classifications: perturbation, differential privacy and generalization.

1) PERTURBATION

Perturbation is commonly used to modify the edge weight values to prevent the edge weights from being utilized for node reidentification while at the same time, maintain the shortest path characteristic between node pairs in the network. A pioneer work was presented in [21], which developed two privacy strategies for different natures of network. The first one is a Gaussian Random Multiplication Perturbation (GRMP) developed for dynamic networks, which adds Gaussian noise to the original edge weights to achieve shortest path preservation. However, the edge weight is power-law distributed in most real life scenarios. Hence, the introduction of Gaussian noise may not guarantee the desired privacy and utility preservation of network data if the edge weights are not normally distributed. The second strategy is a greedy perturbation algorithm developed for static networks. However, it is highly possible for an adversary to reidentify the correct individual by linking the edge weight information to the associated node as some edge weights are unmodified.

A linear programming model was proposed to anonymize the edge weight while preserving the properties of graph that are expressible as linear function of the edge weight [22]. The edge weight is modeled as a matrix and the anonymization is formulated as a linear optimization problem. However, there can be no feasible solution to the optimization problem for large systems which demerits the practicability of this method in scalable social networks.

A k-anonymous path privacy model was presented to protect the sensitive shortest path between two nodes in a weighted graph [23]. It prevents the true shortest path from being revealed by ensuring that there exist at least k shortest paths with the same shortest path distance. Thus, this limits the sensitive path disclosure to a maximum probability of $1/k$. [24] extended k-anonymous path model and modify the edge weights by considering network centrality such as PageRank and nodes' degree. As the edge weights can only be modified once, k-anonymous path privacy cannot be guaranteed when multiple node pairs are involved.

k-anonymous path was further improved in [25] with additional background knowledge of nodes' degree on the shortest path. A (k_1, k_2)-shortest path privacy was proposed to ensure that there are at least k_1 indistinguishable shortest paths between the source and target nodes. In addition, for the non-overlapping nodes on the k_1 shortest paths, there exist at least k_2 nodes with same node's degree and lie on more than one shortest path. There are more restrictions on the modification of edge weight, which lead to a greater information loss than that in [23].

The work of [26] and [52] were combined to propose a k-weighted-degree anonymous model [27]. The edge weights and nodes' degree were assumed as an adversary's background knowledge. This model ensures that in the anonymized graph, there are at least k indistinguishable nodes having the same degree and the distance between the weight sequence of those nodes is within a predefined constant. After obtaining a new degree sequence that is k-degree anonymous using the proposed algorithm in [52], new edge weight values are assigned to the new created edges. The edge weights are adjusted using a linear programming model based on three distance functions (absolute distance, relative distance and rate distance) to ensure that the edge weights generated are nearly valued to other edge weights associated to the node.

2) DIFFERENTIAL PRIVACY

Differential privacy is a relatively new approach to modify edge weight data by adding Laplace noise. It guarantees that the statistical properties of a database is insensitive on a record change. Thus, the output probability of the same results will not change significantly, whether a record is in the data set or not. [28] deployed differential privacy to preserve the privacy of social recommendation. It first clusters the nodes into supergroups, then Laplace noise is added to the average edge weight of each supergroup to modify all edge weights.

In [29], differential privacy was applied to protect the edge weights of social networks and preserve shortest path. The scheme assumed edge weight sequence as an unattributed histogram. Barrels with the same count are merged into one group to reduce the amount of injected noise. Then, Laplace noise is added to edge weight to guarantee

k-indistinguishability between groups so that the number of groups with the same amount of barrels is at least k.

A Variational Bayes-Weighted Network Differential Privacy (VB-WNDP) scheme was proposed with consideration of the structural role [30]. VB-WNDP establishes a probability model of weighted network through Variational Bayes. Noises are added to the parameters of the probability model instead of the edge weights to enhance the data accuracy.

Differential privacy is a strong privacy model as it makes no assumption about the background knowledge of any potential adversary. However, it generates inaccurate results to queries with high sensitivity (for example, kurtosis and correlation). Furthermore, the original data could be estimated with high accuracy from repeated queries.

3) GENERALIZATION

A generalization approach was deployed in [31], where the edge weights are recalculated as the ratio per total edge weight. Particularly, the new edge weight provides very little information about the original network. [32] adopted generalization to generalize the edge weights in an edge group into a range of values. For example, if edge weights 3, 4, 8 and 10 are categorized into a group, then range of values [3,10] is reassigned to the four edge weights. The larger the range, the higher the information loss.

4) OVERALL DISCUSSION ON EDGE WEIGHT ANONYMIZATION

While anonymity has been addressed in the schemes presented, the aspect of unlinkability has not been considered. The schemes discussed do not consider the weight linkability property of network data as the association rules between the original value and the published value are retained in the released data. Hence, the published data leak some useful information of a user and the noise injected could be estimated, provided the association rules are clearly defined to an adversary. **Our work fills the gap of the literature by addressing unlinkability in a social network.** Unlinkability requires that an adversary cannot sufficiently infer the association between the background knowledge of an adversary and the sensitive information of a user. Therefore, no auxiliary edge weight data could be utilized to infer the original edge weight data and the identity of a user with high probability.

From the utility aspect, the aforementioned work were not designed to preserve the statistical properties of original data such as the distribution, mean and standard deviation. **Our work adds to the design of a new edge weight anonymization scheme that fully preserves the statistical properties of a data set based on the idea of data swapping.**

IV. EDGE WEIGHT UNLINKABILITY AND NODE UNLINKABILITY

In this section, we present the definition of some key terms and notation used in this work. We then define *edge weight unlinkability* and *node unlinkability* as two new privacy models in weighted social networks. *Edge weight unlinkability* prevents the inference of true edge weights of a user while *node unlinkability* prevents the linkability of edge weight information to its associated users in the original data.

A. NOTATION

TABLE 1. Notation.

Symbol	Meaning
m	Number of edges in the network
n	Number of nodes in the network
$\boldsymbol{W}$	Weight sequence (Sequence of weight in ascending order)
$\boldsymbol{W'}$	Perturbed weight sequence
$\boldsymbol{W(a)}$	Set of edge weights associated with node a
$\boldsymbol{W(a \cup b)}$	Set of edge weights associated with node a and node b
$W(a,b)$	Edge weight from node a to node b
$W'(a,b)$	Perturbed edge weight from node a to node b
w_p	Edge weight in weight sequence for p =1, 2, 3, ..., m
Z_T	Universal set (Set of distinct values of $\boldsymbol{W}$)
$N(Z_T)$	Complete frequency set (Set of frequency of distinct values in $\boldsymbol{W}$)
Z_p	Possible set (Set of values that satisfy *edge weight unlinkability*)
$N(Z_p)$	Frequency set (Set of frequency of distinct values in possible set)
$\boldsymbol{S}(a,b)$	Candidate set (Set of values that satisfy *node unlinkability*)
N	Number of distinct edge weight values
m_{Add}	Number of fake edges added
n_{Add}	Number of fake nodes added

B. EDGE WEIGHT UNLINKABILITY

We define *edge weight unlinkability* as below.

Definition 1 (***Edge Weight Unlinkability***): Given an edge weight $w \in \boldsymbol{W}$ with value X in an original network G, w is said to be unlinkable if w is perturbed to w' with value Y in a published network G', where $X \neq Y$ and there does not exist an injective function: $f(Y) \mapsto X$ that maps value Y in the published data to value X in the original data. An anonymized data is said to be edge weight unlinkable if all edge weights in the perturbed network G' satisfy *edge weight unlinkability* such that the perturbed edge weight value does not equal to the original edge weight value for all edge weights in weight sequence and there does not exist an injective function f between the original and published data. In mathematical notation, $w'_p \neq w_p, \forall w_p \in \boldsymbol{W}, \forall w'_p \in \boldsymbol{W'}, \forall p = 1, 2, 3, \ldots, m$ and $f(Y) \mapsto X$ is not an injective function.

Here, we provide the proof that *edge weight unlinkability* addresses edge weight disclosure.

Proposition 1: *Suppose an adversary possesses full access to a published data that satisfy edge weight unlinkability, the adversary cannot infer the true edge weights of an arbitrary node in the published data with high probability.*

Proof: From the definition of *edge weight unlinkability*, the mapping function f between the original data and the published data is not injective. This implies that $w_1 \neq w_2$ when $f(w_1) = f(w_2)$. That is, different original edge weight values are mapped to the same published edge weight value. Furthermore, $w'_p \neq w_p$ implies that w'_p could

be selected from $\boldsymbol{W}$ - $\{w_p\}$. Hence, an adversary cannot sufficiently infer the relationship between the original data and the published data as the association rule is not well-defined. In real world scenario, the size of $\boldsymbol{W}$ - $\{w_p\}$ is large. This prevents an adversary from making defined estimation on the original edge weight with high probability. This completes the proof. □

We further evaluate the probability of edge weight disclosure in section VII.

C. NODE UNLINKABILITY

We define *node unlinkability* as below.

Definition 2 (***Node Unlinkability***): Given a node a with associated edge weight sequence $\boldsymbol{W(a)}$ in G and $\boldsymbol{W'(a)}$ in G', the node is said to be unlinkable if $\forall w \in \boldsymbol{W(a)} \Leftrightarrow \nexists\, w \in \boldsymbol{W'(a)}$ and there does not exist an injective function f mapping an original value X to a new value Y. An anonymized data is said to be node unlinkable if all nodes in G' satisfy *node unlinkability* such that $\forall v \in \boldsymbol{V} \wedge \forall w \in \boldsymbol{W(v)} \Leftrightarrow \nexists\, w \in \boldsymbol{W'(v)}$ and there does not exist an injective function: $f(Y) \mapsto X$ that maps value Y in the published data to value X in the original data.

The edge weight data are modified such that the associations between the edge weight values and its nodes are broken. *Node unlinkability* implies *edge weight unlinkability* but not vice versa. The proof is direct from the definition and is omitted. Hence, *node unlinkability* addresses edge weight disclosure.

Here, we prove that *node unlinkability* addresses identity disclosure that relies on edge weight as background knowledge. Particularly, we prove that there does not exist a mapping function that links associated edge weights to its corresponding node in the perturbed data as shown in proposition 2. Moreover, no linkage attack is possible to reidentify a target node in the published data with high probability using edge weight information as background knowledge, as proven in proposition 3.

Proposition 2: *Given there exists a function g that maps a set of edge weights, $\boldsymbol{W(a)}$ to a node a in an original data, such function g does not exist in a perturbed data that satisfy node unlinkability.*

Proof: We prove by contradiction. Given that $\forall w \in \boldsymbol{W(a)}$ are associated (mapped) to a node $a \in \boldsymbol{V}$, we have a function g such that $g(w) \mapsto a$. The existence of the function g indicates that node a is associated with some edge weights w. First, we assume that such function g exists in the perturbed data $\boldsymbol{W'(a)}$. However, based on the definition of *node unlinkability*, $\forall w \in \boldsymbol{W(a)} \Rightarrow \forall w \notin \boldsymbol{W'(a)}$, we know that there does not exist a function g that maps $w \in \boldsymbol{W(a)}$ to the node a in the perturbed data as all the associated edge weights of node a are modified such that $w \notin \boldsymbol{W'(a)}$. Here, we have arrived at a contradiction where our original assumption (function g exists in a perturbed data that satisfy *node unlinkability*) could not be true. This completes the proof. □

Proposition 3: *Given an adversary possesses a complete edge weight information of a known target node a that exists in the network, the adversary fails to reidentify correctly node a in the published data that satisfy node unlinkability using a linkage attack.*

Proof: There are only three possible outcomes of the reidentification. Let b denotes as an arbitrary node in the network and $\boldsymbol{W'(b)}$ is the associated edge weight of b that are published.
Outcome 1: There is no exact match of $\boldsymbol{W(a)}$ and $\boldsymbol{W'(b)}$.
Thus, $\forall a, b \in \boldsymbol{V} \ni \boldsymbol{W(a)} \neq \boldsymbol{W'(b)}$.
∴ No identity is inferred from the published data.
Outcome 2: There is at least one exact match of $\boldsymbol{W(a)}$ and $\boldsymbol{W'(b)}$. We have $\forall a, b \in \boldsymbol{V} \ni \exists\, \boldsymbol{W'(b)} = \boldsymbol{W(a)}$.
From the definition of *node unlinkability*, $\boldsymbol{W(a)} \neq \boldsymbol{W'(a)}$. This implies that $\boldsymbol{W'(b)} \neq \boldsymbol{W'(a)}$.
However, it can be deduced that:
$a = b \Rightarrow \boldsymbol{W(a)} = \boldsymbol{W(b)} \Rightarrow \boldsymbol{W'(a)} = \boldsymbol{W'(b)}$.
Hence, $\boldsymbol{W'(b)} \neq \boldsymbol{W'(a)} \Rightarrow b \neq a$.
∴ Although there is an exact match, a is not the true identity of node b.
Outcome 3: There is at least one partial match of $\boldsymbol{W(a)}$ and $\boldsymbol{W'(b)}$. Thus, $\forall w \in \boldsymbol{W(a)}, \forall w' \in \boldsymbol{W'(b)} \Rightarrow \exists w = w'$.
However, from *node unlinkability*, we have $\forall w \in \boldsymbol{W(a)} \Rightarrow \forall w \notin \boldsymbol{W'(a)}$, which implies that w must not be an edge weight of node a in the published data.
Hence, if $w \in \boldsymbol{W(a)}$ is an edge weight of node b in the published graph, then node a and b must not be the same individual.
∴ Node a cannot be reidentified by linking the edge weight information to the published data.

Therefore, although an adversary possesses a complete edge weight data of a known target node a, the adversary fails to correctly reidentify node a from the published data using a linkage attack. This completes the proof. □

We further evaluate the probability of identity disclosure in section VII.

V. MinSwap

In this section, we design *MinSwap* which deploys *edge weight unlinkability* model to address *edge weight disclosure*. This scheme consists of edge weight modification via data swapping to preserve the edge weight distribution and therefore its statistical properties.

A. MinSwap ALGORITHM

MinSwap consists of two main phases, namely possible set determination and candidate selection. The edge weight data is perturbed by exchanging edge weight values among data tuples to achieve privacy preservation. Data swapping is a value-invariant method where the edge weight distribution is not changed during program execution, only the edge weight sequence is altered. It preserves the univariate statistics such as mean, variance, distribution and lower-order multivariate statistics such as covariance reasonably. A pseudo algorithm of *MinSwap* is presented in Algorithm 1.

Algorithm 1 Minimal Swapping Strategy (*MinSwap*)

Input: The original edge weight sequence, $\boldsymbol{W}$

Output: The perturbed edge weight sequence, $\boldsymbol{W}'$

1 Find Z_T and $N(Z_T)$.
2 **for** p from 1 to m,
3 {Find Z_p and $N(Z_p)$.
4 **if** $N(Z_p) \neq \emptyset$, **then**
5 {Calculate $Prox(w)$ for each $w \in Z_p$.
6 Determine max of $Prox(w)$.
7 Find corresponding w.
8 Update $N(Z_T)$. }
9 **else**
10 {Select a value w from Z_p randomly.
11 Record w in $U(Z_T)$. }
12 Assign the value w to w'_p.}
13 **return** $\boldsymbol{W}'$.

Possible set determination (line 1-3 in Algorithm 1): During the first phase, possible candidates that satisfy *edge weight unlinkability* are determined from the original data. We denote Z_T as the universal set containing all distinct values of $\boldsymbol{W}$ and $N(Z_T)$ as the complete frequency set recording the frequency of values in $\boldsymbol{W}$. The set Z_T is separated into $Z_p \cup \{w_p\}$. The new edge weight (qualified candidate) is selected from the possible set Z_p to ensure that the anonymized data satisfy *edge weight unlinkability*.

Candidate selection (line 5-7 in Algorithm 1): The new edge weight, w' is selected from Z_p based on the maximum of the proximity function ($Prox(w)$), which we define as:

$$Prox(w) = \frac{\text{Frequency of } w \text{ in } Z_p}{|w_p - w|}, \forall w \in Z_p \qquad (1)$$

This function serves two purposes: it allows a nearer value to be selected (a lower information loss) and over the iterations in greedy Algorithm 1, one value could be mapped to different new values (injective function does not exist). This increases the uncertainty of an adversary in inferring the original edge weight value.

Example: An example is demonstrated using data in Figure 2. The original data, $\boldsymbol{W}$, Z_T and $N(Z_T)$ are shown in Table 2. At first iteration, the possible set Z_1 for $w_1 = 1$ is {-, 2, 4, 8, 10, 12, 14, 15} and the frequency set $N(Z_1)$ is {-, 1, 1, 2, 4, 1, 1, 1} (which is obtained by referring the corresponding frequency of each $w \in Z_1$ in $N(Z_T)$). Hence, the new edge weight w'_1 is 2, according to the corresponding maximum of $Prox(w)$. The frequency of 2 is reduced by 1 in the $N(Z_T)$. At the end of algorithm, the final $N(Z_T) = 0$ shows that all the original data are inter-swapped with each other and thus the distribution is fully preserved.

Special Case (line 10-11 in Algorithm 1): $N(Z_p) = \emptyset$ implies that all frequency of values from the possible set are completely consumed. In this case, w' is selected from Z_p randomly, imposing a certain amount of distortion to the original data distribution. However, randomness is applied to provide a higher privacy protection. $U(Z_T)$ is utilized to record the frequency of the overused w. This scenario only occurs when there is a dominant value in the original data ($> 50\%$ of the edge weight data). Edge weight data are big data with high diversity, which ensure the availability of Z_T. **Therefore, the existence of a solution for Algorithm 1 is guaranteed, regardless of the types of distribution of the original data.**

TABLE 2. Example of *MinSwap*.

p	$\boldsymbol{W}$	$N(Z_p)$								$\boldsymbol{W}'$
Z_T		1	2	4	8	10	12	14	15	
$N(Z_T)$		1	1	1	2	4	1	1	1	
1	1	-	1	1	2	4	1	1	1	2
2	2	1	-	1	2	4	1	1	1	1
3	4	0	0	-	2	4	1	1	1	10
4	8	0	0	1	-	3	1	1	1	10
5	8	0	0	1	-	2	1	1	1	10
6	10	0	0	1	2	-	1	1	1	8
7	10	0	0	1	1	-	1	1	1	8
8	10	0	0	1	0	-	1	1	1	12
9	10	0	0	1	0	-	0	1	1	14
10	12	0	0	1	0	1	-	0	1	10
11	14	0	0	1	0	0	0	-	1	15
12	15	0	0	1	0	0	0	0	-	4
Final $N(Z_T)$		0	0	0	0	0	0	0	0	

B. DISCUSSION

It is not highly possible to reverse-engineer and discover the true edge weight as there does not exist an injective mapping between the published data and the original data. The association rules between the original data and the published data are not well-defined. From the utility aspect, the statistical properties of edge weight data are highly preserved as the anonymized data is a permutated version of the original data. This is a scheme designed for networks where the identity of nodes are public knowledge but the edge weight values are sensitive. No node anonymization is required and more utility could be preserved. Examples include research communities (ResearchGate and DBLP) and professional sites (LinkedIn and JobStreet).

VI. δ-MinSwapX

In this section, we design another scheme based on *node unlinkability* to address edge weight disclosure, link disclosure and identity disclosure simultaneously. This scheme consists of edge weight modification using perturbation and structural modification using randomization.

A. EDGE WEIGHT MODIFICATION

Perturbation is deployed to prevent edge weight disclosure and node reidentification using edge weight data as the background knowledge. It consists of two main phases, namely candidate set determination and minimal candidate selection.

Candidate set determination (Algorithm 2): The universal set that contains all the edge weight values (Z_T) is separated

into two mutually exclusive sets, namely candidate set ($\boldsymbol{S}$) and associated edge weight set ($\boldsymbol{W(a \cup b)}$). Candidate set is the set that collects all the possible candidates, such that the candidate $s \in \boldsymbol{S}$ is not associated with node a and b. Candidate set is given by $\boldsymbol{S}(a, b) = \{s | s \in Z_T \text{ - } \boldsymbol{W(a \cup b)} \} = Z_T \setminus \boldsymbol{W(a \cup b)}$. This is to ensure that $\boldsymbol{S}$ contains all the qualified candidates that satisfy *edge weight unlinkability* and *node unlinkability*, as shown in proposition 4.

Algorithm 2 Candidate Set Determination

1 Find universal set Z_T.
2 Find set $\boldsymbol{W(a)} = \{W(a,b) \mid \forall\, a,b \in [1,n]\}$.
3 Find set $\boldsymbol{W(a \cup b)} = \boldsymbol{W(a)} \cup \boldsymbol{W(b)}$.
4 Find candidate set, $\boldsymbol{S} = \{s | s \in Z_T - \boldsymbol{W(a \cup b)} \}$.

Minimal candidate selection (line 5 in Algorithm 3): Candidate is selected based on the least value change to guarantee minimum information loss, as shown in proposition 5. The new edge weight is computed as $w'_p = \min |s - w_p| + w_p$, for $\forall s \in \boldsymbol{S}$.

Algorithm 3 Edge Weight Modification

Input: The original edge weight data, *W(a,b)*

Output: The perturbed edge weight data, *W′(a,b)*

1 Determine the weight sequence, $\boldsymbol{W}$.
2 Find candidate sets for all edge weights.
3 **for** $p = 1$ to m
4 {**call** algorithm 2 to determine the candidate set, $\boldsymbol{S}$.
5 Assign $w'_p = \min |s - w_p| + w_p$, for $\forall s \in \boldsymbol{S}$. }
6 **return** $\boldsymbol{W'}$.

Proposition 4: *Anonymized edge weight data post-implementation of Algorithm 3 satisfy node unlinkability.*

Proof: From the definition 2, we have
$\forall \boldsymbol{w} \in \boldsymbol{W(a)} \Rightarrow \nexists \boldsymbol{w} \in \boldsymbol{W'(a)} \Rightarrow \forall \boldsymbol{w} \notin \boldsymbol{W'(a)}$
Given that $Z_T = \boldsymbol{S} \cup \boldsymbol{W(a \cup b)}$, this implies
$\forall \boldsymbol{w} \in \boldsymbol{W(a)} \Rightarrow \forall \boldsymbol{w} \notin \boldsymbol{S}$
Since the new edge weight is selected from $\boldsymbol{S}$ only, we have $\boldsymbol{w'} \in \boldsymbol{W'(a)} \subseteq \boldsymbol{S}$, which means that $\forall \boldsymbol{w} \notin \boldsymbol{S} \Rightarrow \forall \boldsymbol{w} \notin \boldsymbol{W'(a)}$.
$\therefore \forall \boldsymbol{w} \in \boldsymbol{W(a)} \Rightarrow \forall \boldsymbol{w} \notin \boldsymbol{W'(a)}$
Hence, *node unlinkability* is satisfied, which further implies *edge weight unlinkability*. This completes the proof. □

Proposition 5: *The information loss due to Algorithm 3 is minimum.*

Proof: The information loss occurs during minimal candidate selection. At each iteration, the information loss is $|w'_p - w_p|$. This is the noise injected. The total information loss is $\sum_{p=1}^{m} |w'_p - w_p|$, where m is the number of original data. Since w'_p is selected based on the lowest value change ($\min |s - w_p|$), the total information loss due to Algorithm 3 is minimum. This completes the proof. □

TABLE 3. An example of Algorithm 3.

p	$\boldsymbol{W}$	Value	$\boldsymbol{W(a \cup b)}$ (value)	$\boldsymbol{S}$ (value)	$\boldsymbol{W'}$
1	$W(2,4)$	1	1, 4, 8, 10, 14, 15	2, 12	2
2	$W(6,7)$	2	2, 8, 12, 14, 15	1, 4, 10	1
3	$W(1,2)$	4	1, 4, 8, 10, 14	2, 12, 15	2
4	$W(2,8)$	8	1, 4, 8, 10, 12, 14	2, 15	2
5	$W(3,7)$	8	2, 8, 10, 15	1, 4, 12, 14	4
6	$W(5,8)$	10	8, 10, 12	1, 2, 4, 14, 15	14
7	$W(1,4)$	10	1, 4, 10, 15	2, 8, 12, 14	8
8	$W(2,5)$	10	1, 4, 8, 10, 14	2, 12, 15	12
9	$W(3,8)$	10	8, 10, 12	1, 2, 4, 14, 15	14
10	$W(6,8)$	12	2, 8, 10, 12, 14	1, 4, 15	15
11	$W(2,6)$	14	1, 2, 4, 8, 10, 12, 14	15	15
12	$W(4,7)$	15	1, 2, 8, 10 ,15	4, 12, 14	14

Example: Using the same data set from Figure 2, an example is demonstrated using Algorithm 3 in Table 3. At first iteration, $\boldsymbol{W(2 \cup 4)} = \boldsymbol{W(2)} \cup \boldsymbol{W(4)} = \{1, 4, 8, 10, 14\} \cup \{1, 10, 15\} = \{1, 4, 8, 10, 14, 15\}$. Hence, $\boldsymbol{S} = \{1, 2, 4, 8, 10, 12, 14, 15\} \setminus \boldsymbol{W(2 \cup 4)} = \{2, 12\}$ and $w'_1 = (2 - 1) + 1 = 2$. The iterations terminate at $p = m = 12$.

Discussion: The perturbed data satisfy both *edge weight unlinkability* and *node unlinkability*. A user could not be retraced using edge weight data of the targeted victim as the associations between the edge weights and the nodes have been broken completely. From the utility perspective, we have minimally changed the data so that no excessive utility is loss due to the edge weight modification. If there does not exist a candidate set for a particular edge weight, then no new edge weight is published for that particular edge weight to secure the privacy of a user. However, this is not common in a scalable network which contains high diversity of edge weight values.

B. STRUCTURAL MODIFICATION

Randomization is deployed to modify the network structure to prevent node reidentification using structural data as background knowledge and to prevent link disclosure. It consists of four phases, namely edge deletion, fake node addition, fake edge addition and fake edge weight addition. A pseudo algorithm for structural modification is presented in Algorithm 4.

Edge deletion from existing edges (line 1-5): Most of the prior work modified the graph based on the network centrality of nodes, which measures the influence of the nodes in a graph [24], [27], [31], [34]. In our work, the graph is modified based on the edge betweenness, which represents the importance of an edge in a graph. **Edge betweenness** is the number of shortest paths between pairs of nodes that run along an edge. An edge should not be removed if the edge is important in the network (high edge betweenness). A user-defined parameter δ is selected to remove δ of the existing edges in the ascending order of edge betweenness. A checker $\boldsymbol{C}$ is defined to record the change of structural information. If an edge has been removed, the associated nodes would be removed from $\boldsymbol{C}$.

Algorithm 4 Structural Modification

Input: The perturbed edge weight data, $W'(a,b)$

Output: Perturbed data that resist edge weight disclosure, link disclosure and identity disclosure

1 Define a parameter, δ, where $0 \leq \delta \leq 1$.
2 **while** $\delta \neq 0$,
3 {Edge betweenness is calculated for each edge using original edge weight data. Denotes $\boldsymbol{C}$ as a checker set containing all nodes in the network.
4 Remove δ of the existing edges according to the ascending order of edge betweenness.
5 Record the edge (a, b) that has been removed.

* Remove the corresponding nodes a and b from $\boldsymbol{C}$.

6 Add n_{Add} fake nodes d into the network.

* $n_{Add} = max(\lfloor \frac{|C|}{D_{mode}} \rfloor, 1)$, where D_{mode} is mode of degree. If there are at least one mode, choose maximum mode.

7 **while** $\boldsymbol{C} \neq \emptyset$,
8 {Add edges between the remaining nodes c in $\boldsymbol{C}$ and the fake nodes d randomly until $\boldsymbol{C}$ is empty.

* Randomly select D_{mode} of the remaining nodes c from $\boldsymbol{C}$ to form edges with a fake node d.

9 Record the edge (c, d) that has been formed.

* Remove the corresponding nodes c from $\boldsymbol{C}$.

}
10 **for** each inserted fake edge, assign an edge weight value w' to the edge,
11 {**if** $\exists w \in \boldsymbol{S}(c) \ni w > max[\boldsymbol{W}(c)]$, **then** $w' = min\ w$.
12 **else** $w' = max\ [\boldsymbol{S}(c)]$. } }
13 **return** perturbed data.

Fake node addition (line 6): Some fake nodes d are added into the network to conceal the existence of a target victim in the published data. We determine the minimum number of fake nodes required to be added, n_{Add} as follows:

$$n_{Add} = max(\lfloor \frac{|C|}{D_{mode}} \rfloor, 1) \quad (2)$$

After line 5 of Algorithm 4, $|\boldsymbol{C}|$ represents the number of nodes with intact structural information. Hence, fake edges are formed to modify the structural information of the intact nodes. By considering the degree mode, D_{mode} (degree that appears most often) in original network, all the fake nodes are likely to possess approximately the same degree as the majority nodes in the network (the presence of fake node is hidden). Furthermore, important nodes are preserved in the anonymized network as no true node is removed from the network.

Fake edge addition from non-existing edges (line 7-9): D_{mode} of the remaining nodes c are selected randomly from $\boldsymbol{C}$ to form edges with a fake node d until $\boldsymbol{C}$ is empty. An empty $\boldsymbol{C}$ indicates that all nodes' structural information have been changed. Due to the randomness property of the newly added edges, an adversary could not confidently infer the structural properties of the target victim from the published graph. Furthermore, the structure of the graph is changed without compromising the important nodes and edges in the original network.

Fake edge weight addition (line 10-12): New weight is inserted to each fake edge, which is selected from candidate set of the original node c so that it satisfies *node unlinkability*, such that $w' \in \boldsymbol{S}(c)$. Furthermore, to minimize the influence of these fake edges on the shortest paths of the original network, the new edge weight must satisfy one of the following conditions:

1) If there exists a set of values such that $w \in \boldsymbol{S}(c) \ni w > max[\boldsymbol{W}(c)]$, then $w' = min\ w$.
2) Else, $w' = max[\boldsymbol{S}(c)]$.

***δ-MinSwapX* algorithm:** The pseudo algorithm of *δ-MinSwapX* is a combination of Algorithm 2, 3 and 4. Follow from Table 3, Figure 2 and 5 show the network before and after edge weight modification while Figure 6, 7 and 8 show the network representation after each phase in structural modification, using $\delta = 0.25$.

Discussion: The overall edge modification algorithm is flexible and random. During the edge deletion process, a parameter δ is defined to determine the portion of edges in the network that should be removed. Important edges could be preserved as the edges are deleted according to the influence of edges (edge betweenness). During the edge addition process, the new edges are randomly inserted between the fake nodes and existing nodes in the original network to conceal the true nodes and edges. The δ is used to control the balance between privacy level and utility level. Higher value of δ implies more deletions of true link and thus the probability of link disclosure is reduced. This further implies the larger amount of distortion on the network structure.

Regardless of the value of δ defined, the structural information of all real nodes are modified post-implementation of *δ-MinSwapX*, include degree of node, degree sequence, subgraph and 1-neighbourhood graph of the real nodes. In addition, the edge weight value of the fake edges do not affect the shortest path in original network as the assigned values are slightly larger or equal to the edge

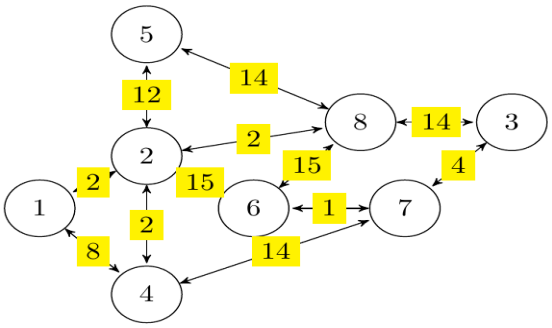

FIGURE 5. Original network after edge weight modification.

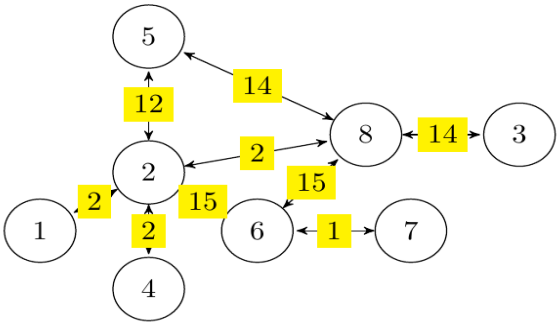

FIGURE 6. Network after edge deletion.

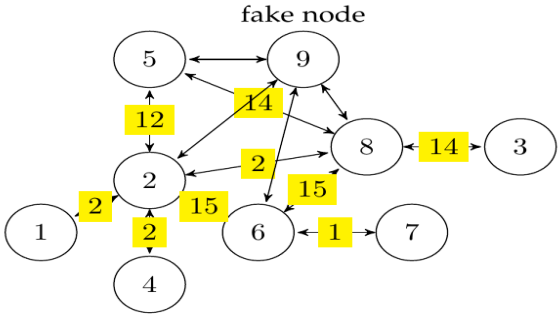

FIGURE 7. Network after fake node and edge addition.

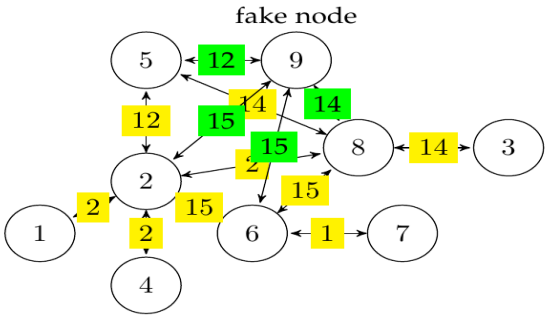

FIGURE 8. Network after fake edge weight addition.

weights involved in that particular shortest path. Hence, the background knowledge of an adversary cannot be utilized to map to the published data for node reidentification as the edge weight and structural information are unlinkable and randomized.

We assume the parameter δ is available to both data miners and attackers [33], [37]. Although δ is known, the identity and link of a user is still protected through the edge randomization process. Note that if $\delta = 1$, the published graph is a null graph (graph with no edge) with $n+1$ nodes, which clearly contains almost no information about the original graph. We intend to have δ to be a small value.

δ-MinSwapX is a scheme designed for networks where the identity, the links and the edge weight data of a user are sensitive information. Edge weight anonymization and structural anonymization are applied simultaneously to fully protect a network user. Examples of such networks include healthcare networks (Doctor On Demand, HelloMD and LiveHealth Online) and social media networks (Facebook, Twitter and Instagram).

VII. SECURITY AND PERFORMANCE ANALYSIS

In this section, we analyze the security level of our schemes theoretically and evaluate the performance of our schemes on three real data sets. All the experiments were conducted on a machine running Microsoft Windows 10 Home Single Language operating system, with an Intel Core TM i7-8750H 2.20 GHz CPU and 16GB RAM. All the algorithms were implemented in Python 3.7.

A. DATA SETS

Three real data sets are used in the experiments to study the performance of our schemes on the data quality in terms of security, efficiency and utility. We extracted a subset of ***Bitcoin Alpha***,[1] ***Facebook Artist***[2] and ***Youtube***[3] to validate the proposed schemes. All the data considered were weighted and non-directed. The details of the data sets are shown in Table 4. The data size is comparable or larger than other relevant work [29], [32], [41]–[43].

B. SECURITY EVALUATION

In this paper, we proposed two schemes that address edge weight disclosure, link disclosure and identity disclosure. We compare our work with some related literature discussed in section III in terms of the privacy components and summarize the comparisons in Table 5. We further analyze the privacy level rendered by our work in proposition 1, 2, 3, 6, 7 and 8.

In the previous work as shown in Table 5, anonymity of edge weight is achieved through the process of data perturbation, k-anonymization, differential privacy and generalization, such that an edge weight could not be reidentified with high probability. As shown in our gap analysis, these schemes do not provide unlinkability feature to the edge weight data. In contrast, our schemes provide anonymity and unlinkability, such that there does not exist an injective mapping between the original and the published data. The edge weight protec-

[1] http://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html
[2] https://snap.stanford.edu/data/gemsec-Facebook.html
[3] https://snap.stanford.edu/data/com-Youtube.html

TABLE 4. Description of the data sets.

Data Set	Nodes	Edges	Details
Bitcoin Alpha	3320	10554	Bitcoin is a peer-to-peer payment system without central authority. Bitcoin Alpha is a platform network which allows users to trade using Bitcoin. In this network, Bitcoin users are anonymous, but users' reputation score are required to reflect the reliability of the traders. Nodes represent the traders, edges are bitcoin transactions and edge weights are rating towards other traders.
Facebook Artist	50515	819306	Facebook is an online social network which allows its users to comment, share photo, post links, chat live and watch video. The data represent mutual like network among verified Facebook pages of artist category and were collected in 2017. Nodes represent the pages, edges are mutual likes among them and edge weights are the number of mutual likes.
Youtube	368548	1048572	Youtube is a video-sharing network, where users represent nodes and they can form friendship with other users in a group. Edge weights are the number of mutual likes.

TABLE 5. Comparison of privacy protection.

Privacy Preservation	Edge Weight Disclosure		Link Disclosure	Identity Disclosure	
	A	U		A	U
[21]–[24], [26], [29], [30]	✔	✖	✖	✖	✖
[33]–[35]	✖	✖	✔	✖	✖
[51]–[57]	✖	✖	✖	✔	✖
[36]–[49]	✖	✖	✔	✔	✖
[25], [27], [28], [31], [32], [50]	✔	✖	✖	✔	✖
MinSwap	✔	✔	✖	✔*	✖
δ-MinSwapX	✔	✔	✔	✔	✔

* A is anonymity, U is unlinkability and ✔* indicates partially addressed.

tion rendered in our schemes is higher since the distinct values in network data are diverse, as shown in proposition 6. All the edge weights are modified in *MinSwap*, providing a certain amount of node protection to the users.

In addition, *δ-MinSwapX* is proposed to provide additional link and node protection. Randomization is deployed to randomly modify the structural information according to the edge betweenness. Random fake edge addition hides the presence of true link in the published graph, and thus prevents the link disclosure, regardless of the background knowledge an adversary may possess, as shown in proposition 7. Furthermore, fake node addition hides the true nodes in the published data. The number of fake nodes and fake edges added are affected by the original data itself, which cannot be inferred by an adversary with high confidence level. Since the edges are randomized, the change of structural information is randomized. An adversary cannot simply map the auxiliary structural information to attack the published data to infer the link and identity of a user. Moreover, *node unlinkability* further guarantees that the edge weight information cannot be linked to its corresponding user in the published data, as shown in proposition 3. The probability of identity disclosure is proven in proposition 8.

Proposition 6: *Suppose an adversary possesses full access to a published data that satisfy edge weight unlinkability, the probability of edge weight disclosure,* $P(w_A) = \frac{1}{N-1}$ *for MinSwap and* $\frac{1}{N-|\boldsymbol{W}'(A\cup B)|}$ *for δ-MinSwapX, where* $N =$ *number of distinct edge weight values in* $\boldsymbol{W}$.

Proof: For an edge weight value w, every other edge weight value in the original data has equal chance of being the new edge weight w' of a victim A. Hence, the probability of edge weight disclosure of victim A under *MinSwap*, $P(w_A) = \frac{1}{N-1}$. If an adversary has a high confidence level, ϵ that the true edge weight lies in a set of x values ($x \leq N-1$), then $P(w_A) = \epsilon\frac{1}{x} + (1-\epsilon)\frac{1}{N-1-x}$. An adversary may not have high confidence level regarding the exact original edge weight values. Hence, when x approaches $N-1$, ϵ approaches to 1, and $P(w_A)$ approaches $\frac{1}{N-1}$.

In the case of *δ-MinSwapX*, an adversary learns that the true edge weight $\in [\boldsymbol{W}'(A \cup B)]'$. Hence, $P(w_A) = \frac{1}{N-|\boldsymbol{W}'(A\cup B)|}$. $P(w_A)$ is arbitrary small since N is arbitrarily large in scalable social network data. This completes the proof. □

Proposition 7: *Given the assumption of adversary in section II, the probability of inferring the presence of link under δ-MinSwapX* $= 1-\delta$ *and the probability of reidentification of the true link* $= \frac{(1-\delta)m}{(1-\delta)m+m_{Add}}$, *where* m_{Add} *is the number of fake edges added.*

Proof: The probability of inferring the presence of link $= 1-\delta$, as δ of the original link are removed from the graph under *δ-MinSwapX*.

The probability of link reidentification = fraction of true link in the published data $= \frac{(1-\delta)m}{(1-\delta)m+m_{Add}}$. This is the same privacy level rendered in [34]. This completes the proof. □

Proposition 8: *Given the assumption of adversary in section II, the probability of identity disclosure of node A under δ-MinSwapX* $= max\ [\frac{1}{n+n_{Add}}, \prod_{i=1}^{D_A}\frac{1}{N-|\boldsymbol{W}'(A\cup B)|}, \frac{\sigma(n_2-n_1)+n_1}{n_1 n_2 n_3}]$, *where* n_1 *is the number of edges deleted for node A,* n_2 *is the number of edges added for node A and* n_3 *is the number of nodes with* D_A *in the published data.*

Proof: There are three possible alternatives to reidentify a victim A using edge weight and structural data as background knowledge:

a) Brute-force: Every node in the published data has equal chance of being victim A. Hence, the probability of reidentification of victim A, $P(A) = \frac{1}{n+n_{Add}}$.

b) Reconstruct the original edge weight from the published graph and deploy linkage attack: The probability of inferring all the true edge weights is, *P(All edge weights are true)* $= \prod_{i=1}^{D_A}\frac{1}{N-|\boldsymbol{W}'(A\cup B)|}$. By matching the auxiliary edge weight

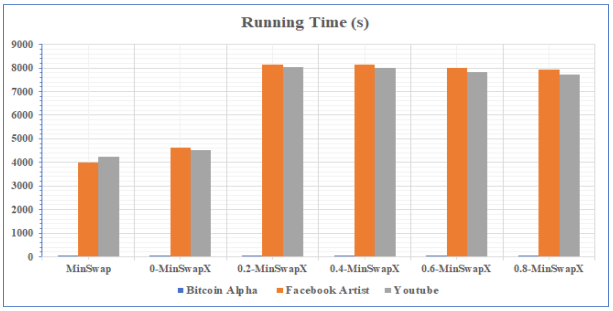

FIGURE 9. Running time (s) according to data sets.

TABLE 6. Comparison of time complexity.

Models	Time Complexity
MinSwap	$\mathcal{O}(m)$
δ-MinSwapX	$\mathcal{O}(n^2 \times log(n) + mn)$
[22], [39]	$\mathcal{O}(n^2)$
[49]	$\mathcal{O}(n \times c \times log_2(T))$
[46]	$\mathcal{O}(n \times h^t)$
[27]	$\mathcal{O}(n^3)$

* m is the number of edges, n is the number of nodes, h is $\frac{2m}{n}$, t is the step for random walk, c is the iteration times and T is the number of groups.

values with the reconstructed edge weights, in the worst case, there is an exact match. $P(A) = \prod_{i=1}^{D_A} \frac{1}{N-|W'(A \cup B)|}$.

c) Reconstruct the original structural graph from the published data and deploy linkage attack: Every node is subjected to either edge deletion or edge addition. The change of degree of node A is $[-n_1, 0) \cup (0, n_2]$. Given an adversary has a confidence level of σ that a node undergoes edge deletion, then the probability of inferring the correct degree, $P(D_A) = \frac{\sigma}{n_1} + \frac{1-\sigma}{n_2}$. If there are n_3 nodes with D_A in the published data, $P(A) = \frac{\sigma(n_2-n_1)+n_1}{n_1 n_2 n_3}$.

This completes the proof. □

C. EFFICIENCY EVALUATION

Figure 9 demonstrates the running time of both *MinSwap* and *δ-MinSwapX* for $\delta = 0$, 0.2, 0.4, 0.6 and 0.8. When $\delta = 0$, only edge weight modification is applied on the data. When $\delta = 1$, a null graph is obtained and hence the time taken is zero. Thus, 1-*MinSwapX* is not considered in the evaluations. The times taken for ***Bitcoin Alpha*** under both schemes are less than 52.5s.

The time complexity (also commonly referred as computational overhead [62]) of *MinSwap* is $\mathcal{O}(m)$, which has a lower time complexity than other models in Table 6. The linear complexity implies the feasibility of *MinSwap* in anonymizing scalable data. The running time increases linearly with the data size. On the other hand, *δ-MinSwapX* has a higher time complexity of $\mathcal{O}(n^2 \times log(n) + mn)$ due to the heavy computation of edge betweenness [63] during the structural modification. Nevertheless, the time complexity of *δ-MinSwapX* is lower than [27] and comparable to [46], [49]. Hence, *δ-MinSwapX* is usable for real world implementation.

D. UTILITY EVALUATION

We study a set of statistical aggregate queries, shortest path analysis and several important graph metrics, which were similarly adopted in [21], [24], [32], [37], [38] to validate the utility of the anonymized graph.

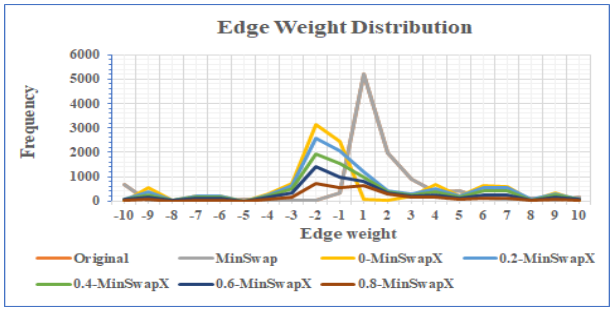

FIGURE 10. Edge weight distribution of ***Bitcoin Alpha***.

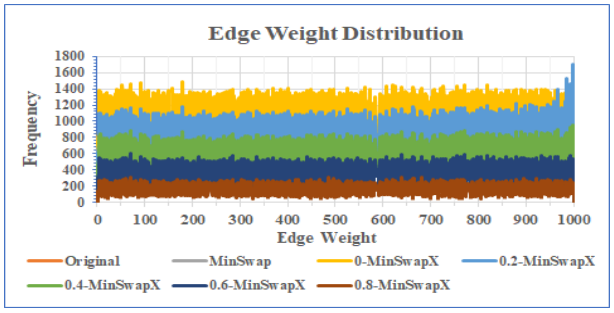

FIGURE 11. Edge weight distribution of ***Facebook Artist***.

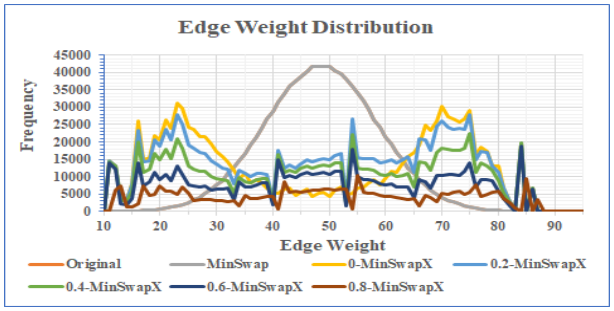

FIGURE 12. Edge weight distribution of ***Youtube***.

1) STATISTICAL ANALYSIS

The impacts of *MinSwap* and *δ-MinSwapX* on the statistical properties of edge weight data are measured using the Kolmogorov-Smirnovb test and statistical aggregate queries. Kolmogorov-Smirnovb test at confidence level = 0.05 is utilized to verify the distribution preservation. As shown in Figure 10, 11 and 12, the distribution of ***Bitcoin Alpha***, ***Facebook Artist*** and ***Youtube*** are preserved at 100% rate under *MinSwap* as all the original data are inter-swapped within the same data set. However, *δ-MinSwapX* does not preserve the distribution of all the three data sets as the edge weight data are modified to satisfy stronger privacy constraints with a minimal utility loss. As the value of δ increases, the level of distortion on the data distribution increases.

Table 7, 8 and 9 show the comparison results of answering statistical aggregate queries. It is observed that *MinSwap* preserves all the statistics at 100% rate as the distributions

TABLE 7. Statistical aggregate query results of ***Bitcoin Alpha***.

Bitcoin Alpha	*Original Data / MinSwap*	*δ-MinSwapX*				
		0	0.2	0.4	0.6	0.8
Mean	1.05	-0.35	-0.03	0.01	0.05	0.25
Median	1	-1	-1	-1	-1	-1
Mode	1	-2	-2	-2	-2	-2
Standard Error	0.03	0.04	0.04	0.05	0.05	0.07
Standard Deviation	3.50	4.32	4.14	4.14	4.06	3.81
Sample Variance	12.22	18.62	17.17	17.15	16.52	14.49
Kurtosis	4.63	-0.14	0.10	0.12	0.26	0.69
Skewness	-1.62	0.39	0.23	0.23	0.19	0.08
Range	20	19	20	20	20	20
Minimum	-10	-9	-10	-10	-10	-10
Maximum	10	10	10	10	10	10

TABLE 8. Statistical aggregate query results of ***Facebook Artist***.

Facebook Artist	*Original Data / MinSwap*	*δ-MinSwapX*				
		0	0.2	0.4	0.6	0.8
Mean	500.53	500.53	510.70	505.78	504.11	503.49
Median	501.00	500.00	514.00	508.00	507.00	505.00
Mode	838.00	179.00	999.00	998.00	69.00	681.00
Standard Error	0.32	0.32	0.35	0.41	0.50	0.70
Standard Deviation	288.82	288.81	292.21	290.13	289.27	289.30
Sample Variance	83415	83411	85386	84177	83674	83695
Kurtosis	-1.20	-1.20	-1.21	-1.21	-1.20	-1.20
Skewness	0.00	0.00	-0.03	-0.02	-0.02	-0.01
Range	999	999	999	999	999	999
Minimum	1	1	1	1	1	1
Maximum	1000	1000	1000	1000	1000	1000

are fully preserved. ***MinSwap* outperforms other existing schemes [21]–[32] in terms of the edge weight statistical properties preservation.** Under *δ-MinSwapX*, most of the query results of each data set remain useful even when the value of δ increases. The maximum deviation of the query results is observed in the mode of ***Facebook Artist*** when δ is 0.6. Although *δ-MinSwapX* is not designed to preserve the statistical properties of the edge weight data, it shows an acceptable preservation rate provided that it guarantees a higher privacy level compared to *MinSwap*. This is a reasonable trade-off between privacy and utility level.

We further analyze the mean absolute error (MAE) of the statistical aggregate query results to measure the average difference between the original data and the published data. The mean absolute error is one of the common statistical metrics which is defined as follows:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{3}$$

where y_i is the simulated result, x_i is the original result and n is the number of observed results. The smaller the MAE,

TABLE 9. Statistical aggregate query results of ***Youtube***.

Youtube	*Original Data / MinSwap*	*δ-MinSwapX*				
		0	0.2	0.4	0.6	0.8
Mean	59.52	59.70	60.35	60.29	60.05	60.02
Median	60.00	60.00	62.00	61.00	61.00	60.00
Mode	58.00	35.00	35.00	87.00	96.00	66.00
Standard Error	0.01	0.02	0.02	0.02	0.03	0.04
Standard Deviation	10.00	23.18	22.11	21.53	21.24	21.31
Sample Variance	100.00	537.12	488.67	463.34	451.03	454.17
Kurtosis	0.00	-1.59	-1.41	-1.27	-1.14	-1.13
Skewness	-0.01	-0.01	-0.08	-0.08	-0.05	-0.04
Range	97	94	93	93	93	93
Minimum	10	13	13	13	13	13
Maximum	107	107	106	106	106	106

TABLE 10. Mean absolute error (MAE) of the statistics.

Statistics	*MinSwap*	*δ-MinSwapX*		
	Bitcoin Alpha/ Facebook Artist/ Youtube	*Bitcoin Alpha*	*Facebook Artist*	*Youtube*
Mean	0.00	1.07	4.39	0.56
Median	0.00	2.00	6.20	0.80
Mode	0.00	3.00	381.20	24.20
Standard Error	0.00	0.02	0.14	0.02
Standard Deviation	0.00	0.60	1.13	11.87
Sample Variance	0.00	4.57	654.82	378.87
Kurtosis	0.00	4.42	0.00	1.30
Skewness	0.00	1.84	0.01	0.05
Range	0.00	0.20	0.00	3.80
Minimum	0.00	0.20	0.00	3.00
Maximum	0.00	0.00	0.00	0.80

the higher the utility of the published data. As shown in Table 10, the MAE of the statistics of all three data sets under *MinSwap* is 0. This implies that there is no difference between the original data and the published data generated by *MinSwap*. On the other hand, Table 10 shows low MAE under *δ-MinSwapX* for all three data sets, except for mode and sample variance of ***Facebook Artist*** and mode, standard deviation and sample variance of ***Youtube***. Nevertheless, the trade-off is affordable and within reasonable bounds as *δ-MinSwapX* assures additional privacy protection compared to *MinSwap*.

2) SHORTEST PATH ANALYSIS

The Dijkstra algorithm is used to determine the shortest paths between all reachable node pairs and evaluate the corresponding shortest path length. We consider the change of shortest path length of the most influential nodes as it is infeasible to evaluate the shortest paths of all reachable nodes in scalable

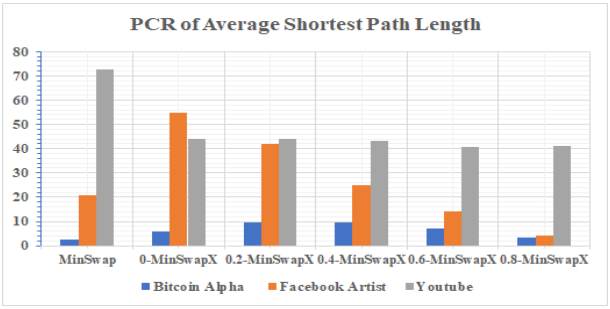

FIGURE 13. Percentage of change per range (PCR) of average shortest path length.

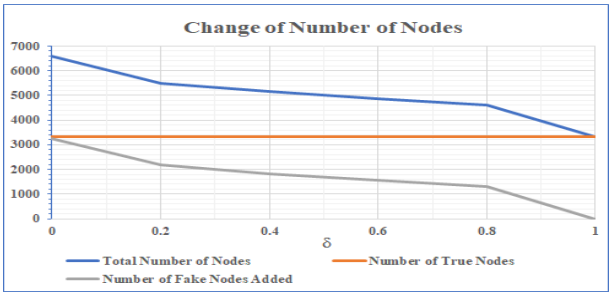

FIGURE 14. Changes of number of nodes in ***Bitcoin Alpha***.

networks. Figure 13 shows the change of average shortest path length.

All data sets show low change of average shortest path length, compared to the range of the edge weight values. This indicates an acceptable preservation rate of average shortest path length rendered by our work.

3) NETWORK CENTRALITY ANALYSIS

We used Cytoscape 3.7.2 as a tool to examine some important graph metrics to evaluate the information loss in δ-*MinSwapX*. *MinSwap* preserves the network structure as no structural modification is applied, and thus is omitted.

The clustering coefficient is a measure of the extent to which nodes in a graph tend to cluster together. The closeness is the inverse of average shortest path length. The normalized connectivity centralization measures the degree to which a graph resembles a star graph topologically. The average degree of a graph is the average number of edges per node in the graph. The diameter of a graph is the maximum distance between all node pairs. The radius of a graph is the minimum among all the maximum distances between a node to all other nodes. The network heterogeneity measures the variance of the degree distribution.

As shown in Figure 14, 15 and 16, the number of fake nodes added decreases as δ increases. Furthermore, the number of fake nodes added is random and depends on the original data itself. All the original nodes are preserved in the published data.

In Figure 17, 18 and 19, as δ increases, the total number of edges, true edges and fake edges added decreases due to the increasing number of edges deleted. Note that the number of

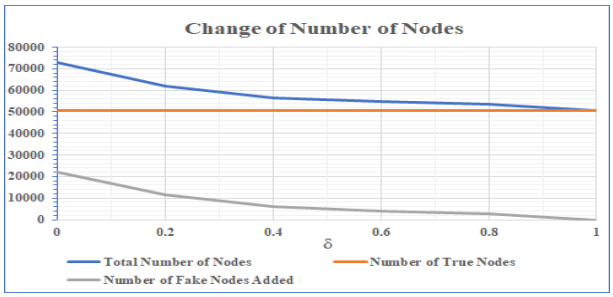

FIGURE 15. Changes of number of nodes in ***Facebook Artist***.

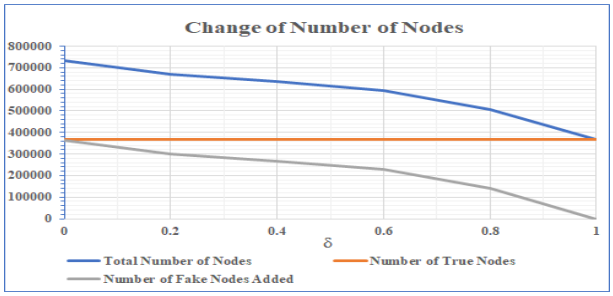

FIGURE 16. Changes of number of nodes in ***Youtube***.

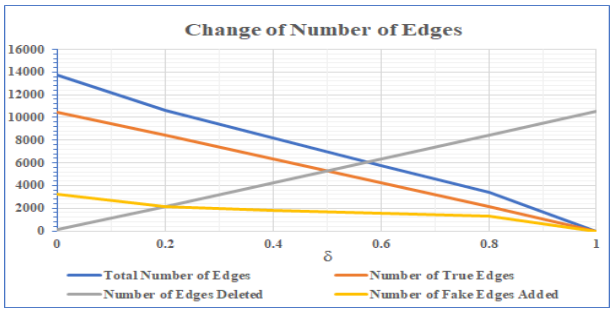

FIGURE 17. Changes of number of edges in ***Bitcoin Alpha***.

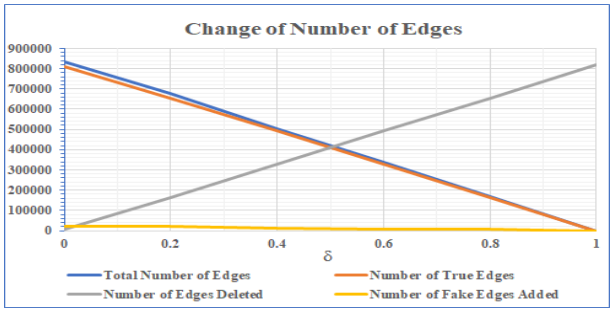

FIGURE 18. Changes of number of edges in ***Facebook Artist***.

fake edges added is random and depends on the data itself. Regardless of the value of δ ($0 < \delta \leq 1$), all the structural information of a node are modified. The higher the value of δ, the larger the amount of edge modification and hence the higher the privacy level rendered.

As δ increases, the edge deletion process compensates the effect of edge addition, which eventually modifies the original graph into a null graph. Therefore, the clustering coefficient, closeness, normalized connectivity centralization and

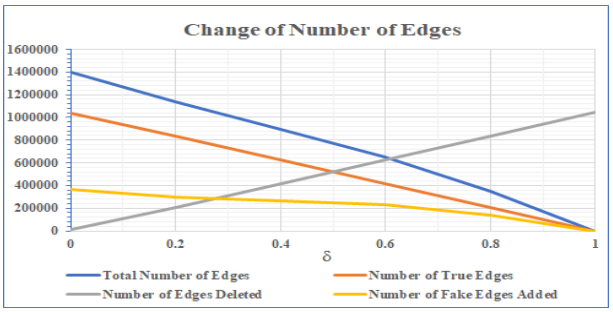

FIGURE 19. Changes of number of edges in ***Youtube***.

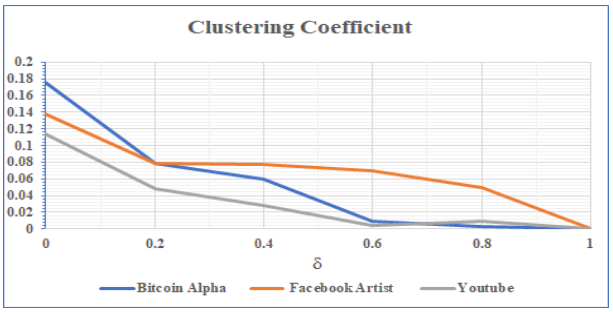

FIGURE 20. Clustering coefficient.

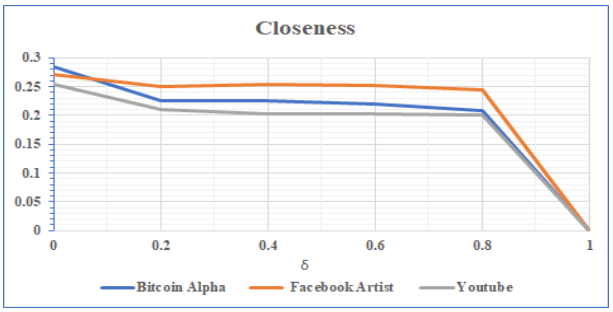

FIGURE 21. Closeness.

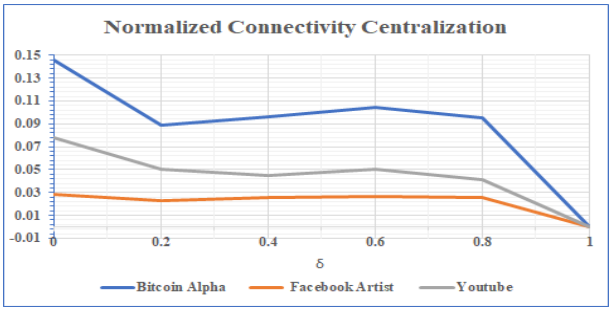

FIGURE 22. Normalized connectivity centralization.

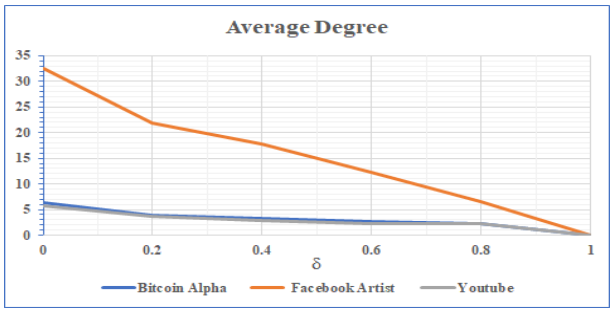

FIGURE 23. Average degree.

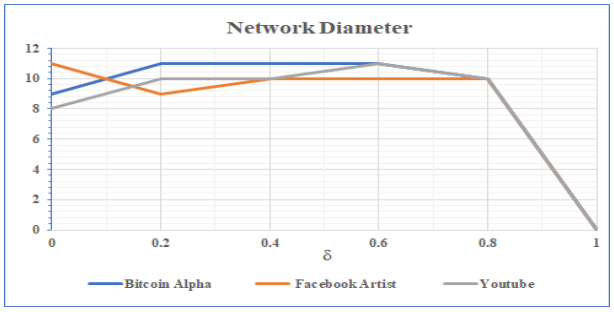

FIGURE 24. Network diameter.

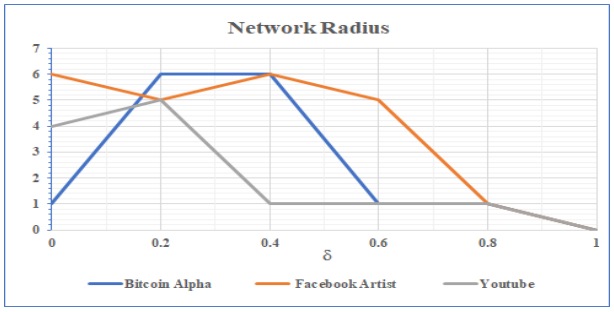

FIGURE 25. Network radius.

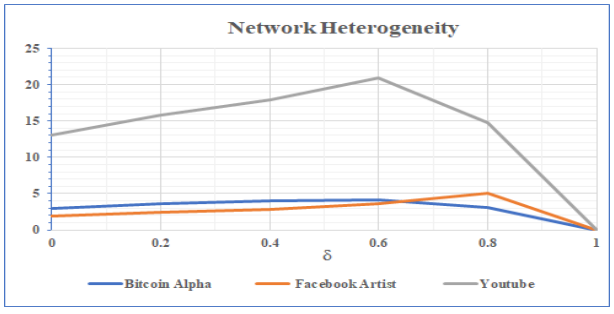

FIGURE 26. Network heterogeneity.

average degree decrease as shown in Figure 20, 21, 22 and 23. Nevertheless, most of the clustering coefficient, closeness, normalized connectivity centralization and average degree remain accurate for $0 < \delta \leq 0.8$.

In Figure 24, the network diameter changes steadily and slightly as δ increases. The network radius of ***Facebook Artist*** is preserved for $0 < \delta \leq 0.6$, as shown in Figure 25. Furthermore, the network radius of ***Bitcoin Alpha*** and ***Youtube*** fluctuates with large magnitude as δ increases. This indicates that the network radius is not preserved in both data sets. In Figure 26, the change of network heterogeneity of all three data sets is consistent and minor over the value of δ. Therefore, the preservation of network centrality is relatively high and δ-*MinSwapX* can be efficiently deployed in scalable real data to provide a high privacy level with a low utility trade-off.

4) DISCUSSION

Most of the existing schemes apply to only unweighted social networks, which do not consider edge weights [33]–[49], [51]–[57]. A weighted graph is a generalization of the unweighted graph. Therefore, our schemes are more practical, such that the proposed schemes provide higher privacy level to the users, in terms of edge weight, link and identity protection by rendering unlinkability in a social network. From the utility aspect, *MinSwap* preserves the statistical properties of edge weight data at rate = 100%. Regardless of the value of δ defined, all the structural information of each node are changed, providing a considerable amount of protections to the users. Furthermore, the average shortest path length and network centrality are well-preserved, considering the degree of privacy protection provided.

VIII. CONCLUSION

In this paper, we studied the problem of privacy-preserving weighted social network data publication. Particularly, our work adds to the design of two secure anonymization schemes based on two new privacy models to efficiently address edge weight disclosure, link disclosure and identity disclosure. *Edge weight unlinkability* and *node unlinkability* are defined to address sensitive edge weight disclosure and node reidentification that rely upon edge weight data as the background knowledge. In addition, edge randomization is deployed to modify the structure of a graph to protect the link and identity of a user against structural attacks. The privacy-preserving ability of our work is evaluated extensively. The empirical study shows that our work maintain high data utility while protect the privacy of users simultaneously.

Overall, we re-emphasize that our work provides the following unique features which are not rendered in other work:

1) Our schemes address three existing privacy problems, namely edge weight disclosure, link disclosure and identity disclosure by achieving anonymity and unlinkability to provide stronger privacy protection.
2) Our schemes efficiently preserve the statistical properties of edge weight data to assure high data utility post-anonymization.
3) Our schemes minimally modify the structural data without eliminating the important edges, and thus resulting in lower information loss compared to other randomization schemes.

For future work, the schemes could be improved for dynamic social networks, where the data are collected and published continuously. Furthermore, another possible direction is to integrate the schemes with differential privacy to further protect the data privacy in an interactive publishing environment.

ACKNOWLEDGMENT

The authors would like to thank Mr. Wong Yik Chun for the assistance in implementing the algorithms into Python 3.7.

REFERENCES

[1] H. Kotani and M. Yokomatsu, "Quantitative evaluation of the roles of community events and artifacts for social network formation: A multilayer network model of a community of practice," *Comput. Math. Org. Theory*, vol. 25, no. 4, pp. 428–436, 2019.
[2] G. C.-C. Shen, J.-S. Chiou, C.-H. Hsiao, C.-H. Wang, and H.-N. Li, "Effective marketing communication via social networking site: The moderating role of the social tie," *J. Bus. Res.*, vol. 69, no. 6, pp. 2265–2270, Jun. 2016.
[3] S. B. Park, C. M. Ok, and B. K. Chae, "Using Twitter data for cruise tourism marketing and research," *J. Travel Tourism Marketing*, vol. 33, no. 6, pp. 885–898, Jul. 2016.
[4] F. Xiong, Y. Liu, and J. Cheng, "Modeling and predicting opinion formation with trust propagation in online social networks," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 44, pp. 513–524, Mar. 2017.
[5] P. Kumbhojkar, M. Jain, E. Rajalakshmi, S. Rawal, and S. Thombre, "Interface implementation for quantifying information spread on social networks," in *Proc. IEEE 6th Int. Conf. MOOCs, Innov. Technol. Educ. (MITE)*, Nov. 2018, pp. 48–51.
[6] M. Burcher and C. Whelan, "Social network analysis as a tool for criminal intelligence: Understanding its potential from the perspectives of intelligence analysts," *Trends Organized Crime*, vol. 21, no. 3, pp. 278–294, Sep. 2018.
[7] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, and C. Piccardi, "Link prediction in criminal networks: A tool for criminal intelligence analysis," *PLoS ONE*, vol. 11, no. 4, pp. 1–21, 2016.
[8] M. Atzmueller, T. Hanika, G. Stumme, R. Schaller, and B. Ludwig, "Social event network analysis: Structure, preferences, and reality," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2016, pp. 613–620.
[9] H. Selim and J. Zhan, "Towards shortest path identification on large networks," *J. Big Data*, vol. 3, no. 1, pp. 1–18, Dec. 2016.
[10] V. Kalavri, T. Simas, and D. Logothetis, "The shortest path is not always a straight line: Leveraging semi-metricity in graph analysis," *Proc. VLDB Endowment*, vol. 9, no. 9, pp. 672–683, 2016.
[11] M. Gong, G. Li, Z. Wang, L. Ma, and D. Tian, "An efficient shortest path approach for social networks based on community structure," *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 114–123, Jan. 2016.
[12] Y. Zhang, Y. Bai, L. Chen, K. Bian, and X. Li, "Influence maximization in messenger-based social networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[13] B. Saoud and A. Moussaoui, "Community detection in networks based on minimum spanning tree and modularity," *Phys. A, Stat. Mech. Appl.*, vol. 460, pp. 230–234, Oct. 2016.
[14] M. A. Ferrag, L. Maglaras, and A. Ahmim, "Privacy-preserving schemes for ad hoc social networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 3015–3045, 4th Quart., 2017.
[15] J. H. Abawajy, M. I. H. Ninggal, and T. Herawan, "Privacy preserving social network data publication," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1974–1997, 3rd Quart., 2016.
[16] J. P. Albrecht, "How the GDPR will change the world," *Eur. Data Protection Law Rev.*, vol. 2, no. 3, pp. 287–289, 2016.
[17] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, 1st ed. Cham, Switzerland: Springer, 2017.
[18] H. N. Chua, S. F. Wong, Y. Chang, and C. F. Libaque-Saenz, "Unveiling the coverage patterns of newspapers on the personal data protection act," *Government Inf. Quart.*, vol. 34, no. 2, pp. 296–306, Apr. 2017.
[19] P. Carey, *Data Protection: A Practical Guide to UK and EU Law*. London, U.K.: Oxford Univ. Press, 2018.
[20] I. Ur Rehman, "Facebook-Cambridge analytica data harvesting: What you need to know," *Library Philosophy Pract.*, no. 2497, pp. 1–11, 2019. [Online]. Available: https://digitalcommons.unl.edu/libphilprac/2497/
[21] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preservation in social networks with sensitive edge weights," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2009, pp. 954–965.
[22] S. Das, O. Egecioglu, and A. El Abbadi, "Anónimos: An LP-based approach for anonymizing weighted social network graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 590–604, Apr. 2012.
[23] S. L. Wang, Z. Z. Tsai, T. P. Hong, and I. H. Ting, "Anonymizing shortest paths on social network graphs," in *Proc. Asian Conf. Intell. Inf. Database Syst.* Berlin, Germany: Springer, 2011, pp. 129–136.
[24] S.-L. Wang, Y.-C. Tsai, H.-Y. Kao, I.-H. Ting, and T.-P. Hong, "Shortest paths anonymization on weighted graphs," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 23, no. 01, pp. 65–79, Feb. 2013.

[25] S. L. Wang, C. C. Shih, I. H. Ting, and T. P. Hong, "Degree anonymization for k-shortest path privacy," in *Proc. IEEE Int. Conf. Syst. Man Cybern.* Manchester, U.K.: IEEE, 2013, pp. 1093–1097.
[26] L. Liu, J. Liu, J. Zhang, and J. Wang, "Privacy preservation of affinities in social networks," in *Proc. Int. Conf. Inf. Syst.*, 2010, pp. 372–376.
[27] M. Yuan and L. Chen, "Node protection in weighted social networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Springer, 2011, pp. 123–137.
[28] L. Chen and P. Zhu, "Preserving the privacy of social recommendation with a differentially private approach," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Dec. 2015, pp. 780–785.
[29] X. Li, J. Yang, Z. Sun, and J. Zhang, "Differential privacy for edge weights in social networks," *Secur. Commun. Netw.*, vol. 2017, pp. 1–10, Mar. 2017.
[30] Y. Wang, J. Yang, and J. Zhang, "Differential privacy for weighted network based on probability model," *IEEE Access*, vol. 8, pp. 80792–80800, 2020.
[31] K. S. Babu, S. K. Jena, J. Hota, and B. Moharana, "Anonymizing social networks: A generalization approach," *Comput. Electr. Eng.*, vol. 39, no. 7, pp. 1947–1961, Oct. 2013.
[32] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1417–1429, May 2017.
[33] X. Ying and X. Wu, "Randomizing social networks: A spectrum preserving approach," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2008, pp. 739–750.
[34] A. M. Fard and K. Wang, "Neighborhood randomization for link privacy in social network analysis," *World Wide Web*, vol. 18, no. 1, pp. 9–32, Jan. 2015.
[35] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Proc. Int. Workshop Privacy Secur. Trust KDD*. Berlin, Germany: Springer, 2007, pp. 153–171.
[36] P. Liu, Y. Bai, L. Wang, and X. Li, "Partial k-anonymity for privacy-preserving social network data publishing," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 27, no. 1, pp. 71–90, Feb. 2017.
[37] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," Dept. Comput. Sci., Fac. Publication Ser., UMass Amherst, Amherst, MA, USA, 2007, pp. 1–17.
[38] J. Cheng, A. W. C. Fu, and J. Liu, "K-isomorphism: Privacy preserving network publication against structural attacks," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2010, pp. 459–470.
[39] P. Liu, L.-E. Wang, and X. Li, "Randomized perturbation for privacy-preserving social network data publishing," in *Proc. IEEE Int. Conf. Big Knowl. (ICBK)*, Aug. 2017, pp. 208–213.
[40] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *Proc. 9th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 169–178.
[41] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 425–438.
[42] P. Liu, Y. Xu, Q. Jiang, Y. Tang, Y. Guo, L.-E. Wang, and X. Li, "Local differential privacy for social network publishing," *Neurocomputing*, vol. 391, pp. 273–279, May 2020.
[43] W. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proc. Int. Conf. Manag. Data*, 2016, pp. 123–138.
[44] K. R. Macwan and S. J. Patel, "Node differential privacy in social graph degree publishing," *Procedia Comput. Sci.*, vol. 143, pp. 786–793, 2018.
[45] J. Su, Y. Cao, and Y. Chen, "Privacy preservation based on key attribute and structure generalization of social network for medical data publication," in *Proc. Int. Conf. Intell. Comput.* Cham, Switzerland: Springer, 2019, pp. 388–399.
[46] H. Zhu, X. Zuo, and M. Xie, "DP-FT: A differential privacy graph generation with field theory for social network data release," *IEEE Access*, vol. 7, pp. 164304–164319, 2019.
[47] F. Ahmed, A. X. Liu, and R. Jin, "Publishing social network graph eigenspectrum with privacy guarantees," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 892–906, Apr. 2020.
[48] T. Gao and F. Li, "Protecting social network with differential privacy under novel graph model," *IEEE Access*, vol. 8, pp. 185276–185289, 2020.
[49] H. Huang, D. Zhang, F. Xiao, K. Wang, J. Gu, and R. Wang, "Privacy-preserving approach PBCN in social network with differential privacy," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 931–945, Jun. 2020.
[50] M. E. Skarkala, M. Maragoudakis, S. Gritzalis, L. Mitrou, H. Toivonen, and P. Moen, "Privacy preservation by k-anonymization of weighted social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2012, pp. 423–428.
[51] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 102–114, Aug. 2008.
[52] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2008, pp. 93–106.
[53] C.-H. Tai, P. S. Yu, D.-N. Yang, and M.-S. Chen, "Privacy-preserving social network publication against friendship attacks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1262–1270.
[54] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proc. IEEE 24th Int. Conf. Data Eng.*, vol. 8, Apr. 2008, pp. 506–515.
[55] B. Zhou and J. Pei, "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowl. Inf. Syst.*, vol. 28, no. 1, pp. 47–77, Jul. 2011.
[56] L. Zou, L. Chen, and M. T. Özsu, "K-automorphism: A general framework for privacy preserving network publication," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 946–957, Aug. 2009.
[57] A. Campan and T. M. Truta, "Data and structural k-anonymity in social networks," in *Proc. Int. Workshop Privacy Secur. Trust KDD*. Berlin, Germany: Springer, 2009, pp. 33–54.
[58] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.
[59] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, 2006, pp. 1–12.
[60] S. Chester, B. M. Kapron, G. Srivastava, and S. Venkatesh, "Complexity of social network anonymization," *Social Netw. Anal. Mining*, vol. 3, no. 2, pp. 151–166, Jun. 2013.
[61] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.
[62] J. Y. Hua, A. Tang, Q. Y. Pan, K. K. R. Choo, H. Ding, and Y. Z. Ren, "Practical–anonymization for collaborative data publishing without trusted third party," *Secur. Commun. Netw.*, vol. 2017, pp. 1–10, Jan. 2017.
[63] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.

KAH MENG CHONG received the B.Sc. degree (Hons.) from the National University of Malaysia, in 2017. He is currently pursuing the M.Sc. degree in mathematics with the University of Malaya. His current research interests include big data, network security, and privacy.

AMIZAH MALIP received the M.Sc. degree in mathematics of cryptography and communications and the Ph.D. degree in information security from the Royal Holloway, University of London, Surrey, U.K. She is currently a Senior Lecturer with the Institute of Mathematical Sciences, University of Malaya, Malaysia. Her main research interests include information security and cryptographic applications.