Federated Graph Analytics with Differential Privacy

Shang Liu
shang@db.soc.i.kyoto-u.ac.jp
Kyoto University
Kyoto, Japan

Yang Cao
yang@ist.hokudai.ac.jp
Hokkaido University
Sapporo, Japan

Masatoshi Yoshikawa
yoshikawa-mas@g.osaka-seikei.ac.jp
Osaka Seikei University
Osaka, Japan

ABSTRACT

Graph analytics across various sources yields valuable insights; however, ensuring privacy becomes increasingly challenging. Federated analytics offers a promising privacy-preserving framework for data analytics. Nonetheless, most current methods are geared towards generic tabular data, limiting their effectiveness in complex graph analytics. In this paper, we first present *federated graph analytics* (FGA), a new paradigm that calculates common graph statistics (i.e., degree distribution, subgraph counting, etc.) across several distributed subgraphs. A key challenge with FGA is the limited view each client has of the global graph, making it challenging for each client to obtain accurate graph statistics. To tackle this, we propose an FGA framework that supports various graph analytics while providing a privacy guarantee. We find that the overlapping information among distributed subgraphs leads to redundant randomization, influencing the utility significantly. To mitigate this issue, we put forward an improved method based on private set intersection (PSI) techniques. By perturbing each item in disjoint subgraphs only once, we reduce the level of added noise considerably. Extensive experiments conducted on real-world graphs demonstrate that our improved method outperforms the baseline by over 70% in most cases.

CCS CONCEPTS

• **Security and privacy → Social aspects of security and privacy**; **Human and societal aspects of security and privacy**.

KEYWORDS

Federated Analytics, Graph Statistics, Differential Privacy

ACM Reference Format:
Shang Liu, Yang Cao, and Masatoshi Yoshikawa. 2023. Federated Graph Analytics with Differential Privacy. In *KDD FL4Data-Mining '23, August 7, 2023, Long Beach, CA, USA.*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Graph data has emerged as a critical resource for modeling and analyzing big data in various applications, including finance, social network, and healthcare. As big data develops, processing or analyzing graph data over multiple parties becomes common. Graph analytics at scale provide valuable insights but they pose more challenges for individual privacy protection.

Federated Analytics (FA) [2, 8, 24], a distributed collaborative analytic paradigm, is a promising privacy-preserving framework for extracting insights across multiple parties without revealing any local information beyond the targeted insights. Over the past decades, the most prominent application of federated analytics is Federated Learning (FL) [3, 16, 27]. Despite similarities, FA and FL are distinct. Most FL works research on training and aggregating the parameters of a specified machine learning model, while FA concentrates on gathering basic statistics from a generalized perspective. Thus, FL can be viewed as a complex FA statistic on the distributed data.

In this paper, we introduce Federated Graph Analytics (FGA), a new computational paradigm for common graph statistics such as degree distribution, subgraph counting, and give some illustrative examples in real-world as follows:

Example 1.1. Federated Social Graph Analytics. Consider that several social platforms (e.g., Facebook, Twitter, LINE, etc.) with their own subgraphs collaboratively analyze the global social network, such as degree distribution [17] (i.e., number of edges connected to a node), subgraph countings (e.g., the number of triangles, stars, or cliques) [9], etc.

Example 1.2. Federated Financial Graph Analytics. Consider that thousands of financial institutions (e.g., banks) collect their own local subgraphs for internal analysis [6]. The demands for graph analytics across distributed subgraphs are increasing. For example, a risk assessment has to be conducted among different banks before lending money to an individual or an enterprise.

Example 1.3. Federated Hospital Graph Analytics. Consider that several hospitals own their private graph about the spread of certain diseases, for instance, COVID-19. It is necessary that the health administration in a certain region asks several hospitals collaborate with each other to detect the spread of COVID-19.

Although privacy-preserving FA models have made considerable progress, supporting graph statistics presents a greater challenge. Earlier FA frameworks for tabular data analytics allowed each client to obtain an intermediate answer based on their own data, which was then sent to an untrusted mediator. But in FGA, each client has a limited view of the global graph and requires information from other clients before obtaining the intermediate answer. For example, as shown in Figure 1, if the query task Q is the triangle counting, each client will return the intermediate answer $Q_i(G_i)$ that is equal to 0. Here, three edges of the targeted triangle $< 2, 3, 4 >$ are from three different clients, which makes it difficult for any client to find any triangle in its local subgraph.

To address the above challenge, we propose a novel FGA framework that supports common graph statistics (i.e., degree distribution, k-stars, triangle counts, etc.) while providing formal privacy

KDD FL4Data-Mining '23, August 7, 2023, Long Beach, CA, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

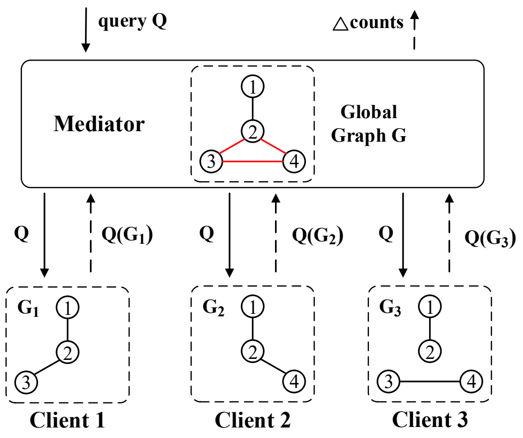

Figure 1: An example of FGA: triangle counting.

guarantee, such as differential privacy protection [7]. To the best of our knowledge, this is the first work to compute graph statistics via FGA. The key idea is that the untrusted mediator collects noisy subgraphs from distributed clients and aggregates a noisy global graph, upon which various graph statistics are then executed. During the collection and aggregation, we use Edge Differential Privacy (Edge DP) [10, 11, 28, 29] to provide a privacy guarantee for each client's sensitive information.

Despite the FGA framework resolving the limited view of the global graph, overlapping information across multiple clients leads to excessive noise. To be specific, for every private collection and aggregation, each client has to randomize each item once. In the worst case, each item needs to be perturbed by m times, and the privacy budget ε has to be divided by m, namely, $\varepsilon_l = \varepsilon/m$, where m is the number of clients. To address this, we propose an improved method based on Private Set Intersection (PSI) techniques [5, 22]. We first generate a disjoint set of distributed subgraphs securely using PSI. Then each silo only randomizes the disjoint subgraph once using privacy budget ε instead of ε/m. The utility is improved significantly since less noise is needed during privacy protection.

The main contributions are as follows:

- We identify the key challenge of federated graph analytics (FGA): a limited view of the global graph makes it difficult for each client to execute graph statistics.
- We introduce an FGA framework to support common graph statistics while providing a formal privacy guarantee.
- We find that overlapping information among different clients results in much utility loss. We propose an improved approach based on PSI technology. Each silo only perturbs each item of its subgraph once, reducing the added noise significantly.
- We conduct extensive experiments to verify the feasibility of our proposed algorithms.

The rest of this paper is structured as follows. Section 2 provides the preliminaries. Section 3 introduces our proposed FGA framework and improved approach. Section 4 presents our experimental results. Section 5 reviews previous works on federated analytics and graph analytics, and Section 6 draws a conclusion.

2 PRELIMINARIES

Federated Graph Data. In this paper, we consider undirected graphs with no additional attributes on nodes or edges within the context of Federated Graph Analytics (FGA). The FGA scenario includes m clients and an untrusted mediator. Each client S_k owns a subgraph $G_k = (V_k, E_k)$, where V is the set of nodes, E_k is the set of edges, and $|V| = n$. Each user i in G_k owns one adjacent bit vector $B_i = \{b_{i1}, b_{i2}, ..., b_{in}\}$, where $b_{ij} = 1$ $(i, j \in [1, n])$ if the edge $(v_i, v_j) \in E$ and $b_{ij} = 0$ otherwise. The subgraph can also be represented as $G_k = \{B_1, ..., B_n\}$. The global graph that is aggregated from distributed subgraphs can be represented as $G = \{G_1, ..., G_m\}$ or $G = (V, E)$, where $V = \bigcup_{k=1}^{m} V_k$ and $E = \bigcup_{k=1}^{m} E_k$. It is worth noting that each client is mutually independent of others and there may exist overlaps among different clients, denoted as $G_{k_1} \cap G_{k_2} \neq \emptyset$, where $k_1 \neq k_2$.

Graph Statistics. The untrusted mediator collects graphs privately from distributed clients and executes common graph statistics Q, such as degree distribution and subgraph counting. The number of adjacent edges for one node i is the node degree d_i, namely, $d_i = \sum_{j=1}^{n} b_{ij}$, where n is the number of users in each client. The server computes the degree sequence $seq(G) = \{d_1, ..., d_n\}$ based on a noisy graph and the degree distribution $dist(G)$ can be easily obtained from $seq(G)$ by counting each degree frequency. A triangle is a closed three-path with three edges and three nodes. A k-star (i.e., 2-star and 3-star) consists of a central node and k neighboring nodes. The triangle and k-star countings are used to compute the number of triangles and k-stars in a graph, respectively.

Private Set Intersection. Private Set Intersection (PSI) is a cryptographic paradigm in multi-party computation that allows n parties to compute the intersection of their data securely without revealing any additional information outside the intersection. Although several two-party PSI protocols [4, 18, 20, 21] have been proposed, multi-party PSI (MPSI) protocols [1, 5, 13, 19] become more suitable for numerous real-world applications. We give more details about MPSI technique in Appendix A.

Graph Differential Privacy. In terms of privacy, we use local differential privacy (LDP) to protect local subgraphs. As a graph consists of nodes and edges, we have two different definitions when LDP is applied to graph data: Node-LDP and Edge-LDP. In this paper, we use Edge-LDP which owns better utility than Node-LDP and is sufficient to protect subgraphs in FGA. The definitions for Node-LDP and Edge-LDP are as follows:

Definition 1 (Node-LDP). A random algorithm M satisfies ε-node LDP, iff for any $i \in [n]$, two adjacent bit vectors B_i and B'_i that differ at most $(n-1)$ bits, and any output $y \in range(M)$,

$$Pr[M(B_i) = y] \leq e^{\varepsilon} Pr[M(B'_i) = y]$$

Definition 2 (Edge-LDP). A random algorithm M satisfies ε-edge LDP, iff for any $i \in [n]$, two adjacent bit vectors B_i and B'_i that differ only in one bit, and any output y $\in range(M)$,

$$Pr[M(B_i) = y] \leq e^{\varepsilon} Pr[M(B'_i) = y]$$

3 METHODOLOGY

In this section, we present an FGA framework and an improved method utilizing PSI technology.

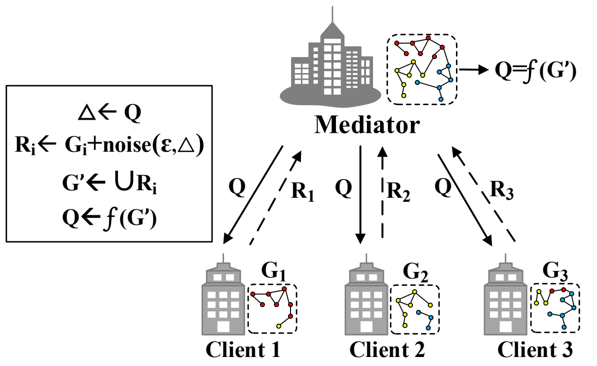

Figure 2: An overview of the FGA framework

3.1 FGA Framework

Initially, we propose an FGA framework for private graph statistics in federated settings. As depicted in Fig. 2, our framework involves several distributed clients and an untrusted mediator. Each client calculates the sensitivity (△) of a given arbitrary graph query function (Q_i), which sets the upper bound on the level of noise each client contributes to maintaining privacy. Next, each client perturbs its local subgraph G_i according to the input privacy budget ε and the sensitivity △. Specifically, given a privacy budget $\varepsilon_l = \varepsilon/m$, every client $k \in [1, m]$ randomizes each bit in its neighbor list with flipping probability $q = \frac{1}{1+e^{\varepsilon_l}}$ using random response (RR) [9, 29]. Following this, the clients send the noisy subgraph to the mediator. Ultimately, the mediator aggregates a noisy global graph and conducts various graph analytics $f(G')$.

The main idea of our FGA framework is shown in Alg. 1. It inputs every client's subgraph, which is represented as adjacent bit vectors $\{B_1, ..., B_n\}$ and the privacy budget ε. Each client first computes its privacy budget, namely, $\varepsilon_l = \varepsilon/m$, where m is the number of clients. Then clients perturb each bit in adjacent bit vectors with a flipping probability $\frac{1}{1+e^{\varepsilon_l}}$, and sends noisy subgraphs to the mediator. The mediator collects and aggregates a noisy global graph and calculates graph statistics upon it.

Empirical Estimation. Applying RR mechanism to an original graph G can lead to a denser problem. Typically, real-world graphs adhere to the power-law distribution and tend to be sparse: an adjacent bit vector contains more 0s than 1s. However, after randomization, the noisy graph G' becomes significantly denser than G: the number of 1s is larger than that of 0s. Here, we employ the known post-processing technique, empirical estimation [9, 23, 29], to achieve unbiased estimations of graph statistics as follows:

Proposition 1. *Let G' be a noisy global graph collected from distributed clients by RR to each adjacent bit vector. Let $d'_i = \sum_{j=1}^{n} b'_{ij}$ be a biased estimation of node i's degree. Let t_0, t_1, t_2, t_3 be the number of no-edges, 1-edges, 2-edges, triangles in G', respectively. Let $f_{d'_i}(G')$ and $f_{\triangle}(G')$ be unbiased estimations of the node degree and triangle counting, respectively. Then*

$$f_{d'_i}(G') = \frac{\sum_{j=1}^{n} b'_{ij}}{2p-1} + \frac{(p-1)n}{2p-1}, p = \frac{e^{\varepsilon_l}}{e^{\varepsilon_l}+1} \tag{1}$$

$$f_{\triangle}(G') = \frac{1}{(e^{\varepsilon_l}-1)^3}(t_3.e^{3\varepsilon_l} - t_2.e^{2\varepsilon_l} + t_1.e^{\varepsilon_l} - t_0) \tag{2}$$

Algorithm 1 FGA Framework

```
Input: Each client's graph $G_k = \{B_1, ..., B_n\}$,
       Privacy budget $\varepsilon$
Output: Target graph statistic $f$
1: Client $S_{k\in[1,m]}$: perturb its graph with $\varepsilon_l = \varepsilon/m$
2: for each $B_i \in G$, where $i \in [1, n]$ do
3:     for each $b_j \in B_i$, where $j \in [1, n], i < j$ do
4:         $b_{j'} = \begin{cases} b_j & w.p.\ \frac{e^{\varepsilon_l}}{1+e^{\varepsilon_l}} \\ 1-b_j & w.p.\ \frac{1}{1+e^{\varepsilon_l}} \end{cases}$
5:     end for
6: end for
7: Client $S_{k\in[1,m]}$: send noisy graph to mediator
8: Mediator: aggregate noisy graph and computes $f$
9: return $f$
```

Algorithm 2 Generate Disjoint Graphs

```
Input: Distributed local graphs $G = \{G_1, ..., G_m\}$
Output: Disjoint local graphs $\hat{G} = \{\hat{G_1}, ..., \hat{G_m}\}$
1: Initialize a set $\hat{G} = \{G_1\}$
2: for each $G_i \in G$, where $i \in [2, m]$ do
3:     $\cap G_i$=MPSI ($G_i$, $\hat{G}$)
4:     $\hat{G_i} = G_i - \cap G_i$
5:     $\hat{G} = \hat{G} \cup \hat{G_i}$
6: end for
7: return $\hat{G}$
```

Limitation of FGA framework. It is noteworthy that overlapping information may exist among different subgraphs, which implies that each sensitive item must be perturbed multiple times (i.e., m times). These multiple perturbations influence the utility significantly. For instance, some users may send money to the same friends or enterprises through different banks. There exists some overlapping information among different financial subgraphs. To protect sensitive information during multiple collections, each sensitive item has to be randomized by m times. Accordingly, the overall privacy budget ε needs to be divided by m, namely, $\varepsilon_l = \varepsilon/m$.

3.2 Improved Method based on PSI

In intuition, if different information of distributed clients is disjoint, each sensitive information only needs to be randomized once and the privacy budget will become $\varepsilon_l = \varepsilon$. In this subsection, we divide the local graphs into multiple disjoint sets privately using private set intersection (PSI). We adopt the SOTA multi-party PSI (MPSI) protocol [12] to compute the intersection privately among multiple clients in semi-honest settings.

The MPSI protocol consists of two main phases: conditional zero-sharing phase and conditional reconstruction phase. Firstly, in the conditional zero-sharing phase, every client S_i securely generates additive sharing of zero for each of its item g_k^i. Here, for easily computing PSI among different clients, we use g_k to represent each edge in G_i and the value of g_k is equal to the sum of source node's ID and destination node's ID. If the other client S_j owns g_k^i, S_i will send the share of zero to it; otherwise, a random value will be sent. In the second phase, each client performs the conditional reconstruction. The idea is that if the sum of shares about item g_k^i is equal to 0, g_k^i will be in the intersection; otherwise, the sum is equal to a random value. The identity privacy of items is guaranteed by a key building block, namely, the oblivious, programmable PRF (OPPRF) [12]. We describe more details about MPSI protocol for graphs in Appendix A.

Based on the MSPI protocol, we propose a method for generating disjoint graphs, as shown in Alg. 2. To be specific, we initialize a virtual set $\hat{G}$ that only includes G_1. Next, each client S_i computes the intersection $\cap G_i$ between G_i and $\hat{G}$. Then we update this virtual set $\hat{G}$ by computing the union between $\hat{G}$ and $(G_i - \cap G_i)$. Finally, it returns a set of m disjoint graphs, namely, $\hat{G}$. Each client randomizes the disjoint graph with the privacy budget ε_l (=ε) as illustrated in FGA framework, and sends a noisy subgraph to the mediator. The mediator aggregates the noisy global graph and finishes various graph statistics.

3.3 Privacy and Security Analysis

THEOREM 3. *Our FGA framework satisfies ε_l-Edge-LDP.*

Proof of Theorem 3: Let $B = \{b_1, ..., b_n\}$ and $B' = \{b'_1, ..., b'_n\}$ are two adjacent vectors that differ in 1 bit. Our randomized algorithm is denoted as M and $Pr[b \rightarrow o]$ denotes the probability that $b \in \{0, 1\}$ becomes $o \in \{0, 1\}$ after the random bit flipping. Let $p = \frac{e^{\varepsilon_l}}{e^{\varepsilon_l}+1}, q = 1 - p$. Given any output $O = (o_1, ..., o_n)$ from M, we have

$$\frac{Pr[M(B) = O]}{Pr[M(B') = O)]} = \frac{Pr[b_1 \rightarrow o_1]...Pr[b_n \rightarrow o_n]}{Pr[r'_1 \rightarrow o_1]...Pr[b'_n \rightarrow o_n]} < \frac{p}{q} = e^{\varepsilon_l}$$

Then according to the parallel composition property and post-processing property [7, 15], we have

$$\frac{Pr[f(G) = Q]}{Pr[f(G') = Q]} < e^{\varepsilon_l}$$

THEOREM 4. *Our improved method satisfies ε-Edge-LDP.*

THEOREM 5. *Our generation method of disjoint subgraphs is secure in the semi-honest model, against any number of corrupting, colluding, semi-honest clients.*

4 EXPERIMENTS

On the basis of the proposed algorithms in Section 3, we would like to answer the following questions in this section:

- For different graph statistics, how much does the improved method via MPSI outperform the basic FGA framework?
- What is the tradeoff between utility and privacy of our proposed methods?

4.1 Experimental Setup

We use two real-world graph datasets from SNAP [14], Facebook graph and Wiki graph, to compare our proposed methods, namely, baseline approach and improved method. The Facebook graph contains 4,039 nodes and 88,234 edges, and Wiki graph owns 7,115 nodes and 103,689 edges. We assume that there are three clients, and for each client's subgraph, we randomly sample a subgraph from each graph dataset with a sampling probability 50%. Then we measure the degree sequence, triangle counting, k-star counting (2-stars and 3-stars). In all experiments, we vary the privacy budget ε from 0.5 to 3. All of our experimental results are the average values computed from 3 runs. We use 'Baseline', Improved' to represent the basic FGA framework and improved method via PSI, respectively. We use the mean squared error (MSE) as a metric to estimate the utility loss under different privacy budgets ε, as used in [29]. In general, the MSE can be computed as $MSE(f(G), f(G)') = \frac{1}{n}\sum_{i=1}^{n}(f(G)_i - f(G)_i')^2$, where n is the number of users in a graph.

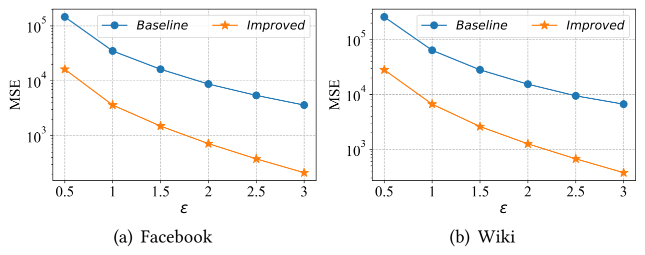

Figure 3: The MSE of degree distribution

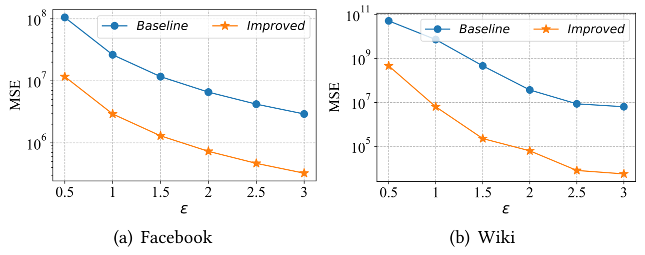

Figure 4: The MSE of triangle counting

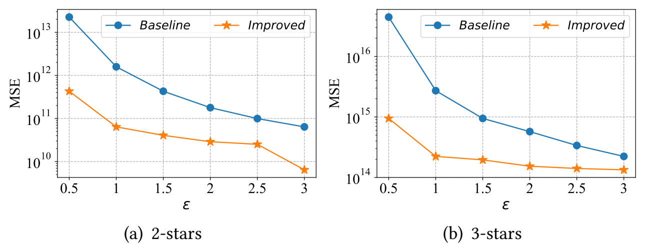

Figure 5: The MSE of k-star counting

4.2 Experimental Results

Fig. 3 shows the results of publishing the degree sequence on Facebook and Wiki graphs. We can find that the improved method outperforms the baseline approach in all cases. To be specific, the MSE of improved method is less than that of baseline by up to 90% on Facebook when $\varepsilon = 3$. The MSE of the baseline is larger than that of improved method by over 80% on Wiki when ε varies from 0.5 to 3. Fig. 4 presents the results of triangle counting on Facebook and Wiki graphs. We can observe that the utility loss of the improved method is much less than that of the baseline in all cases. In particular, the MSE of baseline is larger than that of improved method by over 90% on Wiki graph in most cases. Fig 5 illustrates the results of 2-star and 3-star counting on Facebook graph. We can see that our improved method outperforms the baseline by at least 74.7% on 2-star counting when $\varepsilon = 2.5$. And the MSE of the baseline is larger than that of our improved method by up to 97% when $\varepsilon = 0.5$. Overall, our proposed improved method improves the baseline approach for common graph statistics. The main reason is that we make distributed graphs disjoint using PSI techniques, which leads to only perturbing each item once.

5 RELATED WORKS

The related works are divided into two parts, including federated analytics and graph analytics, which are summarized as follows.

Federated Analytics. The term Federated Analytics was first introduced by Google in 2020 [24], which is explored in support of federated learning for Google engineers to measure the quality of federated learning models against real-world data. The work [2] introduces the notion of Federated Computation that is a means of working with private data at a rather large scale. It provides particular emphasis on federated analytics that is broadly understood as the aspects of federated computation. The authors [26] clarify what federated analytics is and its position in literature, and then present the motivation, application, and opportunities of federated analytics. The paper [8] gives a comprehensive survey about federated analytics. They discuss the unique characteristics of federated analytics and the differences from federated learning. Although there are some previous works studying and discussing the federated analytics recently, they only focus on tabular data which totally differ from graph data in our work.

Graph Analytics. Some works try to process queries over large graphs in distributed user devices. The authors [25] introduce Mycelium for large-scale distributed graph queries with differential privacy. They address three main technical challenges that include topology privacy, neighbor data privacy, and scalability. But the distributed scenario of this paper is different from ours, where each user device only contains a single individual. The paper [30] defines the notion of graph federation for subgraph matching, where graph data sources are temporarily federated. But the proposed framework is too specific for subgraph matching to be useful for other common graph statistics, i.e., degree distribution, subgraph counting, etc. There are a number of research works that publish graph statistics using local differential privacy (LDP), for example, degree distribution [17], subgraph counting [9–11], etc. They belong to a special case of federated graph analytics since each client just includes a single user.

In our work, we consider a general FGA scenario where each client owns an independent subgraph and our proposed methods support various common graph statistics.

6 CONCLUSION

In this work, we have defined a new computational paradigm, Federated Graph Analytics (FGA), which addresses the problem of computing graph statistics with a limited view of the global graph from the perspective of individual clients. We first introduce an FGA framework that can support various common graph statistics while providing a formal privacy guarantee. We find that the overlapping information among clients results in each item being perturbed many times, reducing the utility of the FGA framework. Next, we propose an improved method based on PSI techniques that generates disjoint subgraphs securely. With this approach, each item needs to be randomized only once, improving the overall utility significantly. Comprehensive experimental results verify the utility and privacy achieved by our proposed work.

REFERENCES

[1] Aslı Bay, Zekeriya Erkin, Jaap-Henk Hoepman, Simona Samardjiska, and Jelle Vos. 2021. Practical multi-party private set intersection protocols. *IEEE Transactions on Information Forensics and Security* 17 (2021), 1–15.

[2] Akash Bharadwaj and Graham Cormode. 2022. An Introduction to Federated Computation. In *Proceedings of the 2022 International Conference on Management of Data*. 2448–2451.

[3] Debora Caldarola, Massimiliano Mancini, Fabio Galasso, Marco Ciccone, Emanuele Rodolà, and Barbara Caputo. 2021. Cluster-driven graph federated learning over multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2749–2758.

[4] Andrea Cerulli, Emiliano De Cristofaro, and Claudio Soriente. 2018. Nothing refreshes like a repsi: Reactive private set intersection. In *Applied Cryptography and Network Security: 16th International Conference, ACNS 2018, Leuven, Belgium, July 2-4, 2018, Proceedings*. Springer, 280–300.

[5] Hao Chen, Kim Laine, and Peter Rindal. 2017. Fast private set intersection from homomorphic encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1243–1255.

[6] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. 2022. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition* 121 (2022), 108218.

[7] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[8] Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Shanshan Han, Shantanu Sharma, Chaoyang He, Sharad Mehrotra, Salman Avestimehr, et al. 2023. Federated analytics: A survey. *APSIPA Transactions on Signal and Information Processing* 12, 1 (2023).

[9] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2021. Locally Differentially Private Analysis of Graph Statistics.. In *USENIX Security Symposium*. 983–1000.

[10] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. {Communication-Efficient} Triangle Counting under Local Differential Privacy. In *31st USENIX Security Symposium (USENIX Security 22)*. 537–554.

[11] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. 2022. Differentially Private Triangle and 4-Cycle Counting in the Shuffle Model. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 1505–1519.

[12] Vladimir Kolesnikov, Naor Matania, Benny Pinkas, Mike Rosulek, and Ni Trieu. 2017. Practical multi-party private set intersection from symmetric-key techniques. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1257–1272.

[13] Anunay Kulshrestha and Jonathan Mayer. 2022. Estimating Incidental Collection in Foreign Intelligence Surveillance:{Large-Scale} Multiparty Private Set Intersection with Union and Sum. In *31st USENIX Security Symposium (USENIX Security 22)*. 1705–1722.

[14] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[15] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. 2016. Differential privacy: From theory to practice. *Synthesis Lectures on Information Security, Privacy, & Trust* 8, 4 (2016), 1–138.

[16] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. 2022. Federated graph neural networks: Overview, techniques and challenges. *arXiv preprint arXiv:2202.07256* (2022).

[17] Shang Liu, Yang Cao, Takao Murakami, and Masatoshi Yoshikawa. 2022. A Crypto-Assisted Approach for Publishing Graph Statistics with Node Local Differential Privacy. In *2022 IEEE International Conference on Big Data (Big Data)*. 5765–5774.

[18] Jack PK Ma and Sherman SM Chow. 2022. Secure-Computation-Friendly Private Set Intersection from Oblivious Compact Graph Evaluation. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 1086–1097.

[19] Ofri Nevo, Ni Trieu, and Avishay Yanai. 2021. Simple, fast malicious multiparty private set intersection. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1151–1165.

[20] Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. 2015. Phasing: Private set intersection using permutation-based hashing. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*. 515–530.

[21] Benny Pinkas, Thomas Schneider, Oleksandr Tkachenko, and Avishay Yanai. 2019. Efficient circuit-based PSI with linear communication. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38*. Springer, 122–153.

[22] Benny Pinkas, Thomas Schneider, and Michael Zohner. 2014. Faster private set intersection based on {OT} extension. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 797–812.

[23] Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2017. Generating synthetic decentralized social graphs with local differential privacy. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 425–438.

[24] D. Ramage and S. Mazzocchi. 2020. Federated analytics: Collaborative data science without data collection. https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html.

[25] Edo Roth, Karan Newatia, Yiping Ma, Ke Zhong, Sebastian Angel, and Andreas Haeberlen. 2021. Mycelium: Large-scale distributed graph queries with differential privacy. In *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*. 327–343.

[26] Dan Wang, Siping Shi, Yifei Zhu, and Zhu Han. 2021. Federated analytics: Opportunities and challenges. *IEEE Network* 36, 1 (2021), 151–158.

[27] Zhen Wang, Weirui Kuang, Yuexiang Xie, Liuyi Yao, Yaliang Li, Bolin Ding, and Jingren Zhou. 2022. Federatedscope-gnn: Towards a unified, comprehensive and efficient package for federated graph learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4110–4120.

[28] Chengkun Wei, Shouling Ji, Changchang Liu, Wenzhi Chen, and Ting Wang. 2020. AsgLDP: collecting and generating decentralized attributed graphs with local differential privacy. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3239–3254.

[29] Qingqing Ye, Haibo Hu, Man Ho Au, Xiaofeng Meng, and Xiaokui Xiao. 2020. LF-GDPR: A framework for estimating graph metrics with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (2020), 4905–4920.

[30] Ye Yuan, Delong Ma, Zhenyu Wen, Zhiwei Zhang, and Guoren Wang. 2021. Subgraph matching over graph federation. *Proceedings of the VLDB Endowment* 15, 3 (2021), 437–450.

Algorithm 3 Multi-Party PSI (MPSI) protocol [12]

```
Parameters: $m$ clients $S_1, ..., S_m$.
Input: Client $S_i$ has input $G_i = \{g_1^i, ..., g_t^i\}$
Protocol:
1: For each $i \in [1, m]$ and each $k \in [1, t]$, client $S_i$ chooses random values $\{r_k^{i,j} | j \in [1, m]\}$ subject to $\bigoplus_j r_k^{i,j} = 0$.
2: For each $i, j \in [1, m]$, clients $S_i$ and $S_j$ use an instance of $\mathcal{F}_{OPPRF}^{F,t,t}$ where:
    • $S_i$ is sender with input $\{(g_k^i, r_k^{i,j}) | k \in [1, t]\}$.
    • $S_j$ is receiver with input $G_j$.
  For $g_k^j \in G_j$, let $\hat{r_k}^{i,j}$ represent the output for $g_k^j$ of $F_{OPPRF}$ obtained by $S_j$.
3: For all $i \in [1, m]$ and $k \in [1, t]$, client $S_i$ sets $R_i(x_k^i) = r_k^{i,j} \oplus \bigoplus_{j \neq i} \hat{r_k}^{i,j}$.
4: For each $i \in [2, m]$, client $S_i$ and $S_1$ use an instance of $\mathcal{F}_{OPPRF}^{F,t,t}$ where:
    • $S_i$ is sender with input $\{(g_k^i, R_i(x_k^i)) | k \in [1, t]\}$.
    • $S_1$ is receiver with input $G_1$.
  For $g_k^1 \in G_1$, let $y_k^i$ represent the output for $g_k^1$ of $F_{OPPRF}$ involving $S_i$.
5: Party $S_1$ announces $g_k^1 \in G_1 | R_1(g_k^1) = \bigoplus_{i \neq 1} y_k^i$.
```

A MPSI PROTOCOL

The main protocol is presented in Alg. 3. It consists of two major phases. Firstly, in the conditional zero-sharing phase (step 1-3), each client S_i acts as a dealer for each of its items $g_k^i \in G_i$, and generates a random additive sharing of zero: $\bigoplus_j r_k^{i,j} = 0, j \in [1, m]$. Then each pair of S_i and S_j invoke an instance of oblivious, programmable PRF (OPPRF) (Section 3 in [12]), which is the main building block of MPSI. S_i is a sender with input $\{(g_k^i, r_k^{i,j}) | k \in [1, t]\}$, and S_j acts as a receiver with input G_j. If an item g is shared by all clients, then all pairs of clients will exchange shares securely, and all shares generated a single party XOR to zero. Secondly, in the conditional reconstruction phase (step 4-5), the client S_1 acts as a "dealer" and determines each item $g \in G_1$ whether g is in the intersection. This can also be implemented by an OPPRF. If g is indeed in the intersection, we can obtain $R_1(g) = \bigoplus_{i \neq 1} y^i$; otherwise it will be a random value.