Assessing Centrality Without Knowing Connections

Leyla Roohi(✉), Benjamin I. P. Rubinstein(✉), and Vanessa Teague(✉)

School of Computing and Information Systems, The University of Melbourne, Parkville, VIC, Australia
{Leyla.Roohi,Benjamin.Rubinstein,Vjteague}@unimelb.edu.au

Abstract. We consider the privacy-preserving computation of node influence in distributed social networks, as measured by egocentric betweenness centrality (EBC). Motivated by modern communication networks spanning multiple providers, we show for the first time how multiple mutually-distrusting parties can successfully compute node EBC while revealing only differentially-private information about their internal network connections. A theoretical utility analysis upper bounds a primary source of private EBC error—private release of ego networks—with high probability. Empirical results demonstrate practical applicability with a low 1.07 relative error achievable at strong privacy budget $\epsilon = 0.1$ on a Facebook graph, and insignificant performance degradation as the number of network provider parties grows.

Keywords: Differential privacy · Multi-party computation · Betweenness centrality

1 Introduction

This paper concerns the measurement of node importance in communication networks with *egocentric betweenness centrality* (EBC) [7], representing how much a node's neighbours depend on the node for inter-connection. EBC has emerged as a popular centrality measure that is used widely in practice [12].

EBC computation has many applications. In conjunction with methods for identifying fake news [11,15], EBC can be used to limit its propagation by targeting interventions at those individuals who are most critical in spreading information. EBC computation is straightforward when all communication network information is available to one trusted party. However in reality modern telecommunications involve competing network providers, even within the one country. While many people communicate between countries with completely different networks, where no central authority that can view all their connections. While

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-47436-2_12) contains supplementary material, which is available to authorized users.

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 152–163, 2020.
https://doi.org/10.1007/978-3-030-47436-2_12

recent work [18] considered the case of two mutually-distrusting networks, multiple networks are essential for understanding one person's communication and presents non-trivial technical challenges.

Here we present a protocol that preserves the privacy of the internal connections of each of arbitrarily-many networks while they collaborate on the computation of EBC. By carefully structuring information flow, we achieve highly accurate results and strong privacy protection. We produce a private output that can be safely published. We assume the complete list of nodes (*i.e.*, people) is public, while individual connections are private. Each service provider knows the connections within its own network, plus the connections between one of its members and the outside (*e.g.*, from when they contact someone in a different network). Connections internal to other networks are unknown. We prove that our protocol preserves *edge differential privacy* [8] in which the existence or non-existence of an edge must be protected. We present:

1. A protocol for multi-party private EBC computation;
2. A strengthened adversarial model in comparison to prior work—all participating networks are protected by edge-DP, even when the final output is published;
3. A high-probability utility bound on a core component of our protocol: private distributed release of ego networks;
4. Comprehensive empirical validation on open Facebook, Enron and PGP datasets. This demonstrates applicability of our algorithm with modest 1.07 relative error at strong $\epsilon = 0.1$ DP, practical runtimes, and insignificant degradation with increasingly many parties.

Near-constant accuracy with increasing numbers of parties is both surprising and significant, as is our innovation over past work [18] by preventing leakage at final EBC release. Our protocol is substantially more efficient than a naïve extension of two-party techniques, which would require total communication of $O(|V|^2|A|^2)$ where V is the set of vertices and A the parties. We achieve total communication of $O((|A| + |V|)|A||V|)$. All participants are equal—there is no centralisation; and all parties are protected by edge DP. While we reuse the [18] subset release two-stage sampler, we offer new analysis. We prove in Proposition 1 that it can be distributed without privacy/accuracy loss, and establish a new high-probability utility bound on EBC based on this mechanism's release (Theorem 2).

2 Related Work

In most prior work on differentially-private graph processing, the computation is performed by a trusted authority with complete knowledge of the whole network. Only the output, not the intermediate computations, must preserve privacy [1, 3, 8–10, 16, 17, 19, 20]. There is considerable work on distributed differential privacy [2, 4, 14], where queries are distributed or decomposed into sub-queries. However there is far less work in our distributed privacy model, in which even the intermediate communication should preserve differential privacy. This privacy

model is mostly related to distributed graphs where parties seek joint computation of statistics of the graph. The most closely related work is [18], which derives an edge-DP algorithm for EBC, but only for two networks. The algorithm allows two mutually-distrusting parties to jointly compute the EBC privately using the exponential and Laplace mechanisms to maintain differentially-private intermediate computations and communications. However their work assumes the first party acts as a centraliser that does not share the final result. It is thus not directly applicable to our setting.

3 Preliminaries

3.1 Egocentric Betweenness Centrality

Proposed by [6] as a way of measuring a node's importance by looking only at its *ego network*, the set of nodes directly connected to it. To compute the EBC of node a, we count, for each pair of a-neighbouring nodes (i, j), what fraction of shortest paths between them pass through a. Since we consider only paths within a's ego network, only paths of length two are relevant. We count zero for any pair (i, j) that is directly connected, because these two nodes do not rely on a at all.

Definition 1 ([6]). Egocentric betweenness centrality *of node a in simple undirected graph (V, E) is defined as*

$$\mathrm{EBC}(a) = \sum_{i,j \in N_a : A_{ij}=0, j>i} \frac{1}{A^2_{ij}} \ ,$$

where $N_a = \{v \in V \mid \{v, a\} \in E\}$ denotes the neighbourhood or ego network *of a, A denotes the $(|N_a|+1)\times(|N_a|+1)$ adjacency matrix induced by $N_a \cup \{a\}$ with $A_{ij} = 1$ if $\{i, j\} \in E$ and 0 otherwise; A^2_{ij} denotes the ij-th entry of the matrix square, guaranteed positive for all $i, j \in N_a$ since all such nodes are connected through a.*

3.2 Differential Privacy on Graphs

We wish to protect the privacy of *connections* in networks, which are the edges in the network graph. Differential privacy (DP) [5] limits the differences in an algorithm's output distribution on neighbouring databases, thus quantifying the information leakage about the presence or absence of a particular record. We use *edge privacy* [8], because we wish to control the attacker's ability to make inferences about the presence of individual edges. As graphs on identical node sets, two databases are neighbours if they differ by exactly one edge.

Our databases are simply adjacency matrices in which an element A_{ij} is 1 if there is an edge from i to j and zero otherwise. Equivalently, these can be considered as sequences of bits: elements of $\{0, 1\}^n$ (where n is the number of nodes in the network choose two). Formally, two databases $D, D' \in \{0, 1\}^n$ are termed *neighbouring* (denoted $D \sim D'$) if there exists exactly one $i \in [n]$ such that $D_i \neq D'_i$ and $D_j = D'_j$ for all $j \in [n] \backslash \{i\}$. In other words, $\|D - D'\|_1 = 1$.

Definition 2. *For $\epsilon > 0$, a randomised algorithm on databases or* mechanism *$\mathcal{A}$ is said to preserve* ϵ-differential privacy *if for any two neighbouring databases D, D', and for any measurable set $R \subseteq \text{Range}(\mathcal{A})$,*

$$\Pr\left(\mathcal{A}(D) \in R\right) \leq \exp(\epsilon) \cdot \Pr\left(\mathcal{A}(D') \in R\right) \quad .$$

In this paper, we employ several generic mechanisms from the differential privacy literature including Laplace [5] for releasing numeric vectors, and the exponential [13] for private optimisation. We also use a recent subset release mechanism [18] which leverages the exponential mechanism, and develop its theoretical utility analysis.

Lemma 1 ([18]). *Consider a publicly-known set V and a privacy-sensitive subset $R^\star \subseteq V$. The exponential mechanism run with quality function $q(R^\star, R) = |R \cap R^\star| + |\overline{R \cup R^\star}|$ and $\Delta q = 1$ preserves ϵ-DP. Algorithm 1 (see Supplemental Materials) implements this mechanism, running in $O(|V|)$ space and time.*

4 Problem Statement

We have $|A|$ participating parties $A = \{\alpha_1, \ldots, \alpha_{|A|}\}$, each representing a telecommunications service provider. They control a global communication graph (V, E) whose nodes are partitioned into $|A|$ (disjoint) sets one per service provider s.t. V_α contains the nodes of party α. Every customer is represented as a node that belongs to one and only one service provider; pairs of customers who have had some communication (*e.g.*, a phone call, SMS or email) are edges.

We will often equivalently represent edge sets as adjacency matrices (or flattened vectors) with elements in $\{0, 1\}$.

We write $E_{\alpha,\alpha}$ for the set of edges in (V, E) between nodes in V_α—these are communications that happened entirely within α. Similarly, $E_{\alpha,\beta}$ are the edges with one node in V_α and the other in V_β—these represent communications between two service providers. Set $E = \bigcup_{\alpha,\beta \in A} E_{\alpha,\beta}$ is the disjoint union of all such edge sets.

We assume that all nodes are known to all parties, but that each party learns only about the edges that are incident to a node in its network, including edges within its network.

We wish to enable all parties to learn and publicly release the EBC of any chosen node a, while maintaining *edge privacy* between all parties. Without loss of generality we assume $a \in V_{\alpha_1}$. We also denote by $V^- = V \backslash \{a\}$. Before detailing a protocol for accomplishing this task, we must be precise about a privacy model.

Problem 1 (Private Multi-Party EBC). Consider a simple undirected graph $(\bigcup_{\alpha \in A} V_\alpha, \bigcup_{\alpha,\beta \in A} E_{\alpha,\beta})$ partitioned by parties $A = \{\alpha_1, \ldots, \alpha_{|A|}\}$ as above, and an arbitrary node $a \in V_{\alpha_1}$. The problem of *private multi-party egocentric betweenness centrality* is for the parties α to collaboratively approximate $\text{EBC}(a)$ under assumptions that:

A1. All parties $\alpha \in A$ know the entire node set $\bigcup_{\alpha \in A} V_\alpha$;
A2. Each party $\alpha \in A$ knows every edge incident to nodes within its own network, *i.e.* $\bigcup_{\beta \in A} E_{\alpha,\beta}$.
A3. The computed approximate EBC(a) needs to be available to all of parties.

The intermediate computation must protect ϵ-differential edge privacy of each party from the others. We seek solutions under a fully adversarial privacy model: irrespective of whether other parties follow the protocol, the releases by party α protect its edge differential privacy. (Of course a cheating participant can always release information about edges it already knows, which may join another network.)

Furthermore, the *output* must protect ϵ-differential privacy of the edges. In [18] the final EBC could be revealed only to the party who made the query. In this paper, the final EBC is ϵ-differentially private and can be released safely to anyone.

5 Multi-party Private EBC

We describe three algorithms: SubsetRelease, PrivatePathCount, and PrivateReciprocateandSum, which are privacy-preserving versions of Steps i–iii of Protocol 1 (see Supplemental Materials). These then combine to produce PrivateEBC, a differentially-private version of the whole protocol.

5.1 Private Ego Network Broadcast

Each party α runs SubsetRelease, Algorithm 1 (see Supplemental Material) with its share $R^\star_\alpha$ of a's ego network. It broadcasts the output R_α—the approximation of $R^\star_\alpha$.

SubsetRelease uses the exponential mechanism to privately optimise a particular quality function (Lemma 1) that encourages a large intersection between $R^\star_\alpha$ and release R_α, along with a minimal symmetric set difference. As each party runs this mechanism relative to its own node set, it operates its own quality function defined relative to $R^\star_\alpha$ (see Proposition 1 for the formal definition). We observe a convenient property of the quality functions run by each party: they sum up to the overall quality function if the ego party was to run SubsetRelease in totality. This permits proof (in the Supplemental Materials) that this simple distributed protocol for private ego network approximation exactly implements a centralised approximation. ***There is no loss to privacy or accuracy due to decentralisation.***

Proposition 1. *Consider parties $\alpha \in A$ running* SubsetRelease *with identical budgets $\epsilon_1 > 0$ and quality functions $q_\alpha(R) = |R \cap R^\star \cap V_\alpha| + |\overline{R \cup R^\star} \cap V_\alpha|$, on their disjoint shares $R^\star_\alpha = R^\star \cap V_\alpha$ to produce disjoint private responses $R_\alpha \subseteq V_\alpha$. Then $R = \cup_{\alpha \in A} R_\alpha$ is distributed as* SubsetRelease *run with ϵ, quality function $q(\cdot)$, on the combined $R^\star$ in V. Consequently the individual R_α and the combined R, each preserve ϵ_1-DP simultaneously.*

Algorithm 1. PrivatePathCount

Input: ego node $a \in V_{\alpha_1}$ (remember, by assumption, α_1 contains a); execution party $\alpha \in A$; true node set $R^\star_\alpha$; for each $\beta \in A$, edge set $E_{\alpha,\beta}$ and private node set R_β; $\epsilon_2, \Delta_2 > 0$

Ensure: A vector of noisy counts, indexed by endpoints $\{i, j\}$ with $i < j$, of the total number of nodes k in $R^\star_\alpha$ that are connected to both i and j.

if $\alpha = \alpha_1$ **then**
  $R^\star_\alpha \longleftarrow R^\star_\alpha \cup \{a\}$
end if
$R_A \longleftarrow \bigcup_{\beta \in A} R_\beta$
for $i \in R_A$ **do**
  for $j \in R_A$ with $i < j$ **do**
    $K \longleftarrow \left\{k \in R^\star_\alpha \mid \{i,k\}, \{k,j\} \in \bigcup_{\beta \in A} E_{\alpha,\beta}\right\}$
    $T^{ij}_\alpha \longleftarrow |K| + \mathrm{Lap}(2\Delta_2/\epsilon_2)$
  end for
end for
return $\mathbf{T}_\alpha$

5.2 Private Path Count

Each party α runs Algorithm 1, using the R_β's received from each other party in the previous step. Party α counts all the 2-paths where the intermediate node is in R_α. For each node pair (i, j) with $i < j$, α will send the 2-path count $T^{i,j}_\alpha$ to the party that contains node i, just like the non-private version of the protocol. But first, in order to privatise this vector of counts, Laplace noise is added to the two-path counts according to the sensitivity in the following lemma proved in the Supplemental Materials, thereby preserving ϵ_2-DP in this stage's release.

Lemma 2. *Let query f denote the vector-valued non-private response $\mathbf{T}_\alpha$ of party α in Algorithm 1. The L_1-global sensitivity of f is upper-bounded by $\Delta f = 2|R_A|$.*

5.3 Private Reciprocate and Sum

Every party α receives noisy counts from PrivatePathCount and for any pairs $i < j$ where $i \in R^\star_\alpha$ and $j \in \bigcup_{\beta > \alpha} R_\beta \cup R^\star_\alpha$, that are believed by α to be disconnected, α increments the received $T^{i,j}$ by the number of incident 2-paths. Each party then reciprocates the summation of the counts. In this algorithm, each party may replace noisy R_α with true $R^\star_\alpha$. This optimises utility at no cost to privacy: counts T^{ij} for $i, j \in R_\alpha \backslash R^\star_\alpha$ are discarded. This is safe to do, since the Laplace mechanism already accounts for changes in $R^\star_\alpha$. The Laplace noise is utilised to privatise the reciprocated sum S_α to ϵ_3-DP, calibrated by sensitivity as bounded next, with proof relegated to the Supplemental Materials.

Lemma 3. *Let query f' denote the reciprocate and sum over 2-paths with intermediate point in $\bigcup_\beta R_\beta$ while the nodes $i < j$ are not connected and $i \in R^\star_\alpha$ and*

Algorithm 2. PrivateReciprocateandSum

Input: ego node $a \in V_{\alpha_1}$; execution party $\alpha \in A$; for each $\alpha \leq \beta \in A$, edge set $E_{\alpha,\beta}$ and private node set R_β; for each $\beta \in A$, noisy counts $\mathbf{T}_\beta$; $\epsilon_3, \Delta_3 > 0$

$R \longleftarrow \bigcup_{\beta>\alpha} R_\beta \cup R^\star_\alpha$
$E_\alpha \longleftarrow \bigcup_{\beta\geq\alpha} E_{\alpha,\beta}$
$S_\alpha \longleftarrow 0$
for $i \in R^\star_\alpha$ **do**
  for $j \in R$ with $i < j$ **do**
    if $\{i, j\} \notin E_\alpha$ **then**
      $T \longleftarrow \sum_{\gamma\in A} T^{ij}_\gamma$
      $S_\alpha \longleftarrow S_\alpha + (\lfloor \max\{0, T\} \rfloor + 1)^{-1}$
    end if
  end for
end for
$S_\alpha \longleftarrow S_\alpha + \mathrm{Lap}(2\Delta_3/\epsilon_3)$
return $\mathbf{S}_\alpha$

$j \in \bigcup_{\beta<\alpha} R_\beta \cup R^\star_\alpha$. *Then the* L_1*-global sensitivity of* f' *is upper-bounded by* $\Delta f' = (\lfloor \max\{0, T\} \rfloor + 1)^{-1} \leq 1$ *irrespective of party.*

Communication Complexity. Ego Network Broadcast requires each party to send to each other party $|V|$ bits of length 1 that shows the node is present or not, hence a total of $O(|A|^2|V|)$. Private Path Count sends, for each node i, up to $|V|$ messages $T^{i,j}$ from each party to the owner of node i. The pathcounts T^{ij} are at most $|V|$, so the total size is $O(|A||V|^2)$. Finally, Reciprocate and Sum requires every participant to send each other one message: $O(|A|^2)$. Hence the total communication complexity is $O((|A| + |V|)|A||V|)$.

5.4 PrivateEBC: Putting it All Together

After the parties have run the protocol phases, namely SubsetRelease, PrivatePathCount and PrivateReciprocateandSum, they must finally complete the computation of the private EBC. Algorithm 3 depicts PrivateEBC orchestrating the high-level protocol thus far, and then adding the received S_α to compute final EBC.

Theorem 1. PrivateEBC *preserves* $(\epsilon_1 + \epsilon_2 + \epsilon_3)$*-DP for each party.*

Remark 1. While we have used uniform privacy budgets across parties, our analysis immediately extends to custom party budgets.

6 Utility Bound

In this section we develop a utility analysis of privacy-preserving betweenness centrality, noting that no previous theoretical analysis has been performed including in the two-party case [18]. Our analysis focuses on a utility bound

Algorithm 3. PRIVATEEBC

Input: (Public) ego node $a \in V_{\alpha_1}$; ordered set of parties A; node sets V_α for $\alpha \in A$; parameter vectors $\epsilon, \Delta \succ 0$.
Input: (Private) for each $\alpha, \beta \in A$, edges $E_{\alpha,\beta}$, nodes $R^\star_\alpha$;

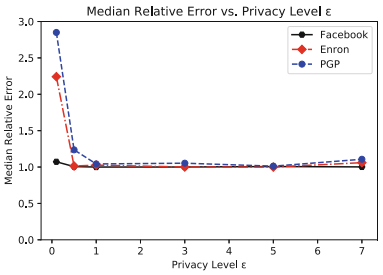
for $\alpha \in A$ in parallel **do**
  Party α does:
  if $\alpha = \alpha_1$ **then**
    $V = V_\alpha \backslash \{a\}$
  else
    $V = V_\alpha$
  end if
  $R_\alpha \leftarrow$ SUBSETRELEASE$(V, R^\star_\alpha, \epsilon_1)$
  Broadcast R_α
  $\mathbf{T}_\alpha \leftarrow$ PRIVATEPATHCOUNT$(\alpha, E_{\alpha\beta}, R_\beta, \epsilon_2, \Delta_2)$
  for all $i, j \in V$ with $i < j$ **do**
    Send $T^{i,j}_\alpha$ to the Party β s.t. $i \in V_\beta$
  end for
  $S_\alpha \leftarrow$ RECIPROCATEANDSUM$(a, \{E_{\alpha,\beta}, R_\beta | \beta > \alpha\}, \epsilon_3)$. {Party α reciprocates and sums only paths with $\beta \geq \alpha$.}
  Broadcast S_α
  pEBC$(a) \leftarrow \sum_{\alpha \in A} S_\alpha$
  Return pEBC(a)
end for

on EBC resulting from the subset release mechanism run to privatise the ego network. We abuse notation with $q(R) = q(R^\star, R)$ referring to the quality function of the SUBSETRELEASE mechanism of Lemma 1 with dependence on the private $R^\star$ made implicit; likewise for the quality functions run by each party in the decentralised setting. The technical challenge is in leveraging the following well-known utility bound on the exponential mechanism, which only establishes high-probability near-optimal quality.

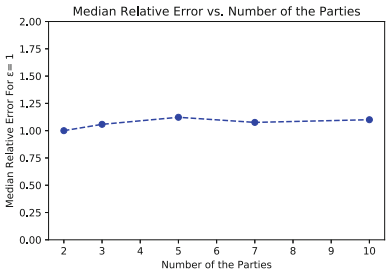
Corollary 1. *Consider parties $\alpha \in A$ each running* SUBSETRELEASE *concurrently with budgets $\epsilon > 0$ and quality functions $q_\alpha(R) = |R \cap R^\star \cap V_\alpha| + |\overline{R \cup R^\star} \cap V_\alpha|$ on their disjoint shares $R^\star_\alpha = R^\star \cap V_\alpha$ to produce disjoint responses $R_\alpha \subseteq V_\alpha$. Then the consequent high-probability quality bound of Lemma 4 (see Supplemental Materials) holds for random combined response $R = \cup_\alpha R_\alpha$.*

We prove the following high-probability utility bound in the supplemental materials.

Theorem 2. *Consider privacy budget $\epsilon > 0$, true ego network $R^\star$ and $t > |R^\star|^2/2$. And suppose that each party $\alpha \in A$ runs* SUBSETRELEASE *with budget ϵ, quality function $q_\alpha(R) = |R \cap R^\star \cap V_\alpha| + |\overline{R \cup R^\star} \cap V_\alpha|$, on their disjoint share $R^\star_\alpha = R^\star \cap V_\alpha$ to produce disjoint private response $R_\alpha \subseteq V_\alpha$. Then EBC_1 produced from $R = \cup_{\alpha \in A} R_\alpha$ incurs error relative to non-private EBC run on non-private $R^\star$, upper bounded as $|EBC - EBC_1| \leq t$, with probability at least:* $1 - \exp\left(-\epsilon(\sqrt{2t} - |R^\star|)/2\right) 2^{|V^-|}$.



(a) Median relative error of the 60 random nodes with ϵ =0.1 to 7, Facebook, Enron and PGP data set, for three parties.



(b) Median Relative error of 120 nodes with $\epsilon = 1$, for different number of parties for PGP.

Fig. 1. Utility of Private EBC for Facebook, Enron and PGP data sets.

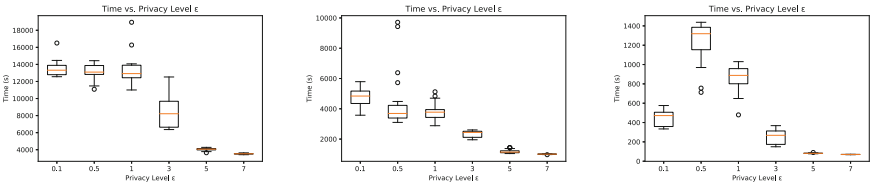
Remark 2. The bound of Theorem 2 can make meaningful predictions (*i.e.*, is non-vacuous). For example **a modest privacy budget of 2.1 is sufficient to guarantee reasonable relative error 3 w.h.p 0.999** for a large ego network spanning half an (otherwise sparse) graph. Similar relative error (for end-to-end private EBC) at similar privacy budgets occurs in experiments on real, non-sparse networks below. Further analysis can be found in the Supplementary Materials.

7 Experimental Setup

In order to validate the utility and privacy of PrivateEBC, we experimented with three different graphs on Facebook friendships with 63,731 nodes and 817,035 edges[1], the Enron email network with 36,692 nodes and 183,831 edges[2] and Pretty Good Privacy (PGP) with 10,680 nodes and 24,316 edges (See footnote 1). We employ uniform random sampling in order to partition the graphs into multiple disjoint parties while keeping the structure of the graph intact. In addition to evaluations on three parties across datasets, we also validated utility across 2, 3, 5, 7 and 10 parties on the PGP data set. The experiments were run on a server with 2×28 core Xeon's (112 threads with hyper threading) and 1.5 TB RAM, using Python 3.7 without parallel computations. We employed the `Mpmath` arbitrary precision library for implementing inverse transform sampling (Algorithm 2 in the supplemental materials) and set the precision to 300 bits. We use relative error between true EBC and private EBC—the lower the relative error the higher the utility. Any errors around 1 or 2 are considered practical as they signify EBCs within the same order of magnitude. We ran the experiment 60 times for each chosen value ϵ by choosing the target ego nodes

[1] Institute of Web Science and Technologies at the University of Koblenz–Landau: The Koblenz network collection (2018).

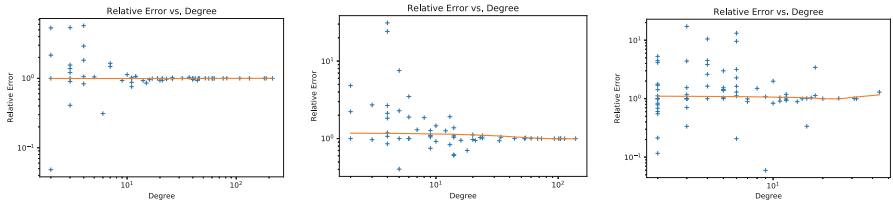
[2] Stanford University: Stanford large network dataset collection (2009).



(a) Time of computing 60 random nodes with ϵ =0.1 to 7, Facebook data , for three parties.

(b) Time of computing 60 random nodes with ϵ =0.1 to 7, Enron data set , for three parties.

(c) Time of computing 60 random nodes with ϵ =0.1 to 7, PGP data set, for three parties.



(d) Relative error of 60 nodes with different degrees for ϵ =1, Facebook data set.

(e) Relative error of 60 nodes with different degrees for ϵ =1, degree, Enron data set.

(f) Relative error of 60 nodes with different degrees for ϵ =1, PGP data set.

Fig. 2. Timing results and effect of degree for Facebook, Enron and PGP data sets.

randomly and robustly aggregating the relative error by median. Throughout we set $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon/3$.

8 Results

First, we demonstrate how PrivateEBC utility varies with increasing privacy budget from 0.1 to 7, for three parties across each of three different graph datasets. The median relative error between real and private EBC represents utility. Figure 1(a) displays the results for Facebook, Enron and PGP data sets, where median relative error decreases significantly when ϵ is increased to a strong guarantee of 1, and remains small for larger ϵ. For strong privacy guarantee of $\epsilon = 0.5$, median relative error is usually ≈ 1 for all three data sets. These results demonstrate that PrivateEBC ***achieves practical utility across a range of graph sizes and privacy levels.*** Next we report utility at privacy $\epsilon = 1$ for the number parties ranging over 2, 3, 5, 7 and 10. Every point in Figure 1(b) shows the median relative error between private and real EBC across 120 randomly chosen nodes in the PGP data set. ***Our results find insignificant degradation occurs to accuracy or privacy when growing the number of parties.***

Remark 3. While more parties means more calls to ReciprocateandSum and PrivatePathCount such that the scale of the second and third mechanisms' Laplace noise increases moderately, the major source of error, SubsetRelease, is not affected by the number of parties as proved in Proposition 1.

We report on timing analysis for PrivateEBC as a function of privacy. Median computation time of 60 random ego nodes for ϵ budget from 0.1 to 7 is reported in Figures 2(a), 2(b) and 2(d), on Facebook, Enron and PGP data sets. Here total time overall decreases as privacy decreases (increasing ϵ), while a small increase to runtime can be seen at very high levels of privacy (low but increasing ϵ) for Enron it is likely due to different behaviours in the protocol with increasing ϵ. When the set difference of R and $R^\star$ is small, the two-stage sampler generates small numbers of nodes in faster time. However faster runtime with lower privacy dominates behaviour overall. Figures 2(d), 2(e), 2(f) show how the median relative error is changing by ego node degree. We report results on privacy budget $\epsilon = 1$, which do not show significant dependence: In Facebook the median relative error is almost constant for different node degrees and in Enron and PGP for node degrees up to 10^2, deviations are approximately 1% and 0.5% of the maximum relative error respectively.

9 Conclusion and Future Work

This paper develops a new protocol for multi-party computation of egocentric betweeness centrality (EBC) under per-party edge differential privacy. We significantly improve on past work by extending to multiple parties, achieving very low communication complexity, theoretical utility analysis, the facility to release the private EBC to all parties. Experimental results demonstrate the practical accuracy and runtime of our protocol at strong levels of privacy.

For future work we hope to allocate differential privacy budgets per stage, by optimising utility bounds. We also intend to develop a network model that reflects a person's use of multiple media, so that the node set need not be disjointly partitioned, while the privacy of edges remains paramount.

References

1. Bhaskar, R., Laxman, S., Smith, A., Thakurta, A.: Discovering frequent patterns in sensitive data. In: SIGKDD 2010, pp. 503–512 (2010). https://doi.org/10.1145/1835804.1835869
2. Chen, R., Reznichenko, A., Francis, P., Gehrke, J.: Towards statistical queries over distributed private user data. In: NSDI, pp. 13–13 (2012)
3. Day, W.Y., Li, N., Lyu, M.: Publishing graph degree distribution with node differential privacy. In: SIGMOD 2016, pp. 123–138 (2016). https://doi.org/10.1145/2882903.2926745
4. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_29

5. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
6. Everett, M., Borgatti, S.P.: Ego network betweenness. SN **27**(1), 31–38 (2005). https://doi.org/10.1016/j.socnet.2004.11.007
7. Goh, K.I., Oh, E., Kahng, B., Kim, D.: Betweenness centrality correlation in social networks. Phys. Rev. E **67**(1), 017101 (2003). https://doi.org/10.1103/physreve.67.017101
8. Hay, M., Li, C., Miklau, G., Jensen, D.: Accurate estimation of the degree distribution of private networks. In: ICDM, pp. 169–178 (2009). https://doi.org/10.1109/icdm.2009.11
9. Karwa, V., Raskhodnikova, S., Smith, A., Yaroslavtsev, G.: Private analysis of graph structure. PVLDB **4**(11), 1146–1157 (2011). https://doi.org/10.1145/2611523
10. Kasiviswanathan, S.P., Nissim, K., Raskhodnikova, S., Smith, A.: Analyzing graphs with node differential privacy. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 457–476. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_26
11. Kwon, S., Cha, M., Jung, K., Chen, W., Wang, Y.: Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th ICDM, pp. 1103–1108. IEEE (2013). https://doi.org/10.1109/icdm.2013.61
12. Marsden, P.V.: Egocentric and sociocentric measures of network centrality. Soc. Netw. **24**(4), 407–422 (2002). https://doi.org/10.1016/s0378-8733(02)00016_3
13. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS 2007, pp. 94–103. IEEE (2007). https://doi.org/10.1109/focs.2007.66
14. Mohammed, N., Alhadidi, D., Fung, B.C., Debbabi, M.: Secure two-party differentially private data release for vertically partitioned data. IEEE Trans. Dependable Secure Comput. **11**(1), 59–71 (2013). https://doi.org/10.1109/tdsc.2013.22
15. Monti, F., Frasca, F., Eynard, D., Mannion, D., Bronstein, M.M.: Fake news detection on social media using geometric deep learning. arXiv preprint arXiv:1902.06673 (2019)
16. Mülle, Y., Clifton, C., Böhm, K.: Privacy-integrated graph clustering through differential privacy. In: EDBT/ICDT Workshops, pp. 247–254 (2015)
17. Raskhodnikova, S., Smith, A.: Efficient Lipschitz extensions for high-dimensional graph statistics and node private degree distributions. arXiv preprint arXiv:1504.07912 (2015)
18. Roohi, L., Rubinstein, B.I.P., Teague, V.: Differentially-private two-party egocentric betweenness centrality. In: The 38th Annual IEEE International Conference on Computer Communications. INFOCOM (2019). https://doi.org/10.1109/infocom.2019.8737405
19. Shen, E., Yu, T.: Mining frequent graph patterns with differential privacy. In: KDD 2013, pp. 545–553 (2013). https://doi.org/10.1145/2487575.2487601
20. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Private release of graph statistics using ladder functions. In: SIGMOD 2015, pp. 731–745 (2015). https://doi.org/10.1145/2723372.2737785