exactly as the BGW protocol implements it; hence, the security of this sub-protocol relies on the security of the BGW protocol [3, 1]. Interpreting the protocol as D sequential invocations of the F_{add}-functionality, we have in fact a protocol in the F_{add}-hybrid model. It is easy to see that in this protocol, the corrupted parties do not receive any messages from the honest parties and their view consists of the outputs of the F_{add} invocations only. However, these values are exactly the vectors $\mathbf{s}_1, \ldots, \mathbf{s}_D$ which the simulator receives as input (as the output of the F_{add}-functionality for the corrupted parties). Therefore, simulation is straightforward, where the simulator just outputs the values $\mathbf{s}_1, \ldots, \mathbf{s}_D$ (that it received as input) as the outputs of F_{add}; clearly, those values are exactly the view of the corrupted parties in the real execution of the protocol Π_4. □

6. EXPERIMENTS

Since our protocols are provably secure (Theorems 2 and 3), and correct (i.e., return the same answer as their non-secure counterparts), the main goal of our experimental assessment is to study their efficiency. In this regard we will report and analyze their total work defined as the sum of running times over all hosts. Observe that, while the number and total size of communication messages may also be of interest, no experiments are required here as these numbers are completely determined by our protocols (see Table 1).

Experiments settings. All our experiments are run on a dual-core Intel i7-5600U CPU (2.60 GHz) with 16Gb RAM under Linux. We implemented our protocols in C++ and compiled the code using g++ with speed optimizations.

We used the NTL library[5] to perform computations with arbitrary length integers, as well as polynomial interpolation over large finite fields (as required by our protocols). The finite fields we use are integers modulo a prime large enough to guarantee exact computation of the results (e.g., of length $O(\log \ell)$ for π_1, and $O(D \log(|V|\ell))$ for π_2 and π_3). The threshold parameter ℓ' is set to $\lfloor(\ell-1)/2\rfloor$ for Π_1, Π_2, and Π_3 as in Theorem 2 (recall that $\ell \geq 3$ denotes the number of hosts). For Π_4 we set $\ell' = \ell$ as this protocol allows the use of the ℓ-out-of-ℓ secret sharing scheme (see Section 5). The weights β_k for matrix powers are set to $1/2^k$, $1 \leq k \leq D$.

Datasets. For our experiments we used publicly-available real-world multigraphs of various types and sizes, spanning different domains: offline relationships (aarhus), road networks (london), genetic interactions (hiv, arabi), online social networks (higgs, obama), interactions in sharing sites (youtube), human genealogies (wikitree) and co-authorship networks (dblp). All except dblp are separated by layers; for dblp we use instead the timestamp information to partition the edges into 3 groups. Some of these graphs are undirected; in this case we duplicate each edge. Table 2 reports their main characteristics and origin.[6,7,8,9]

6.1 π_3 as a median ranking

In this section we show that, if the hosts are interested in a ranking of "influential" users, the ranking obtained by joining forces through secure multiparty computation is intuitively better than the individual rankings that each host should be able to obtain on its own. Specifically, the ranking obtained by π_3 on the multilayer graph lies "in the middle" of all the individual rankings obtained on each layer (π_2 and π_3 coincide when using a single layer).

Table 2: Dataset characteristics: name, directed (D) or undirected (U), source (please refer to the footnotes below), number of hosts, number of nodes, number of edges in the union graph $G = (V, E)$ (corresponding to the sum of entries of the matrix A), and number of edges considering their multiplicity (corresponding to the sum of entries of the matrix B).

dataset	U/D	url	ℓ	$\|V\|$	$\|E\|$	$\sum_{t=1}^{\ell} \|E_t\|$
aarhus	U	6	5	61	353	620
london	U	6	3	369	430	503
hiv	D	6	3	1,005	2,310	2688
arabi	U	6	7	6,980	17,497	18,117
youtube	U	7	5	14,992	10,726,107	32,980,158
higgs	D	6	3	304,691	904,404	1,110,962
wikitree	D	8	4	1,382,750	9,620,090	18,381,878
dblp	U	9	3	1,314,050	10,724,828	18,986,618
obama	D	6	3	2,281,259	6,283,002	9,182,052

Table 3: Kendall-Tau correlations between rankings induced by the 7 individual layers of arabi, and π_3-based ranking using all layers. The last column is the sum of the row values minus 1.

layer	1	2	3	4	5	6	7	π_3	$\sum$ - 1
1	1.0	-0.307	0.042	-0.001	0.033	0.052	0.0	0.314	0.13
2	-0.307	1.0	0.013	0.01	0.018	0.041	0.055	0.452	0.28
3	0.042	0.013	1.0	0.392	0.272	0.028	-0.002	0.055	0.8
4	-0.001	0.01	0.392	1.0	0.237	0.046	0.017	0.023	0.72
5	0.033	0.018	0.272	0.237	1.0	0.053	0.026	0.036	0.68
6	0.052	0.041	0.028	0.046	0.053	1.0	0.14	0.063	0.42
7	0.0	0.055	-0.002	0.017	0.026	0.14	1.0	0.031	0.27
π_3	0.314	0.452	0.055	0.023	0.036	0.063	0.031	1.0	**0.97**

Table 4: Kendall-Tau correlations between rankings induced by the 5 individual layers of youtube, and π_3-based ranking using all layers. The last column is the sum of the row values minus 1.

layer	1	2	3	4	5	π_3	$\sum$ - 1
1	1.0	0.421	0.379	0.386	0.219	0.401	1.8
2	0.421	1.0	0.37	0.239	0.317	0.483	1.83
3	0.379	0.37	1.0	0.35	0.353	0.615	2.07
4	0.386	0.239	0.35	1.0	0.137	0.374	1.49
5	0.219	0.317	0.353	0.137	1.0	0.617	1.64
π_3	0.401	0.483	0.615	0.374	0.617	1.0	**2.49**

We proceed as follows for an ℓ-layer dataset:

(1) Compute all π_3 (or equivalently π_2) scores based on each single layer, obtaining ℓ rankings $R_1, \ldots, R_\ell$. Then we obtain the additional ranking $R_{\ell+1}$ induced by the π_3 scores on the multilayer network.

(2) Compute a symmetric table with the Kendall-Tau[10] correlation between all the rankings from the previous step.

(3) For each $i \in [\ell + 1]$, define the global score of i as the sum of all Kendall-Tau correlations between R_i and R_j for $j \neq i$. The larger this value is, the better the ranking i correlates with all other rankings on average.

Tables 3 and 4 show the correlations thus obtained for arabi and youtube, respectively. We verify that the global score of the ranking induced by π_3 is larger than any of the individual scores. That is, by applying a secure protocol to compute π_3, all parties can obtain a ranking that is closer on average to all individual rankings.

6.2 Efficiency of Π_1, Π_2, and Π_3

We next report runtime comparison of the three basic protocols Π_1, Π_2, and Π_3. We already know, by the computational complexity analysis reported in Section 5, about the inefficiency of Π_1, which constraints us to use small graphs for this comparison.

[5] http://www.shoup.net/ntl/

[6] http://deim.urv.cat/~manlio.dedomenico/data.php

[7] http://socialcomputing.asu.edu/datasets/YouTube

[8] http://proj.ise.bgu.ac.il/sns/wikitree.html

[9] http://konect.uni-koblenz.de/networks/dblp_coauthor

[10] https://en.wikipedia.org/wiki/Kendall_tau_distance