We thank the reviewers for the work. **To Reviewer 1:** **(a)** $\mathcal{H}(G\restriction_{X_j})$ is the result of the $\mathcal{H}$ function on the induced subgraph $G\restriction_{X_j}$ of G. **(b)** As suggested, we tested our results on a variants of modularity QDS as baseline [1]. Fig.(left) confirms that REM still performs better over datasets Jaz and Eml. **(c)** To explain the inferior performance of MOM compared to RAN, we remark that modularity is an index easily affected by community size [2]. MOM will most likely add edges to two communities with the largest volume (See Lem. 2 in Supplementary II). We also gave explanation at the end of Sec. 2 in the main paper. **To Reviewer 2:** **(a)** Due to restriction to use url, we cannot share code that shows detailed implementation. The $O(|V|L)$ complexity can be justified as follows: For communities X_i, X_j, suppose a data structure is used that assigns to each node $x \in X_i$ the node $x' \in X_j$ where no edge exists between x and x', and x' is such a node with minimum degree. To find the desired critical edge, the algorithm may scan over all such pairs (x, x') where $x \in X_i$. This takes $O(|X_i|)$. Similarly, the algorithm examines over all pairs (y, y') where $y \in X_j$ and $y' \in X_i$ is defined analogously as x'. This takes $O(|X_j|)$. Hence, for X_i, X_j the algorithm takes $O(|X_i| + |X_j|)$. Thus, for any X_i, the algorithm will take $O(L|X_i| + |X_1| + \cdots + |X_L|) = O(L|X_i| + |V|)$. The overall time takes $O(L|X_1| + \cdots + L|X_L| + L|V|) = O(L|V|)$. The implementation of the required data structure would store for each node $x \in X_i$, collections of nodes $Y_d, Y_{d+1}, \ldots, Y_g \subseteq X_j$ where Y_d contains all nodes $y \in X_j$ such that $\{x, y\}$ is a non-edge and y has degree d, where d, g are the least and greatest integers where Y_d, Y_g are non-empty. This makes sure that the data structure can be built and updated in the required time complexity. **(b)** Section 3 motivates structural entropy from a data communication perspective, to stay consistent with the origin of information theory. Our entropy notion measures the cost of communication between nodes in the network that is inherent to network topology. We view each node as a communication station that is able to pass message to adjacent nodes. To send a message, the sender needs to give an "address" of the receiver node, which is a string in a fixed (say binary) alphabet. This is the so-called *codeword*, which is used to locate a node in the network. The structural entropy $\mathcal{H}(G)$ is the expected length of the codeword when we assume that the messages are passing between randomly chosen node to a randomly chosen neighbor. **(c)** "Lossless way" refers to a data compression algorithm that allows the original data to be perfectly reconstructed from the compressed data. **(d)** The size of dataset Dol is very small with only 62 vertices. Adding 100 random edges will drastically change the original community structure which explains the good performance of RAN. **To Reviewer 3**. **(a)** REM aims at minimizing $\rho_{\mathcal{P}}(G)$. See Fig.(right) as an example of edge adding which distorts the community structure of a graph. To measure how much privacy is leaked, we use the normalized mutual information D between structure (a) and (b). Thus we may view $1 - D$ as an indication of how much sensitive network structure is protected. **(b)** Affiliation relation disclosure can lead to serious privacy leaking. E.g., Wondracek et al. in [3] showed that information about the community memberships of a user (i.e., the groups of a social network to which a user belongs) is sufficient to uniquely identify this person, or, at least, to significantly reduce the set of possible candidates. In [4], communities are used to re-identify multiple addresses belonging to a same user in Bitcoin trading networks. **(c)** By adding a fixed (small) number of edges, we aim to minimize the change to other "insensitive" information of a network. The table lists changes to the clustering coefficient, mean shortest path length, the percentage of nodes with the top-10% Pagerank and Betweenness after applying our algorithm.

Dataset	$\lvert E' \rvert$	Jaccard	Clustering coefficient	Mean shortest path length	10% Pagerank	10% Betweenness
Dol	10	$1 \to 0.44$	$0.308 \to 0.298$	$3.357 \to 2.996$	$1 \to 0.833$	$1 \to 0.833$
Jaz	250	$1 \to 0.48$	$0.520 \to 0.498$	$2.23 \to 2.070$	$1 \to 0.895$	$1 \to 0.842$
Eml	100	$1 \to 0.39$	$0.166 \to 0.166$	$3.606 \to 3.577$	$1 \to 0.991$	$1 \to 0.982$
PGP	400	$1 \to 0.45$	$0.378 \to 0.377$	$7.485 \to 7.279$	$1 \to 0.979$	$1 \to 0.930$
CAI	1000	$1 \to 0.49$	$0.007 \to 0.007$	$3.875 \to 3.869$	$1 \to 0.983$	$1 \to 0.977$
Bri	1000	$1 \to 0.44$	$0.111 \to 0.111$	$4.858 \to 4.854$	$1 \to 0.993$	$1 \to 0.971$

[1] Chen, Kuzmin, Szymanski. Community Detection via Maximization of Modularity and Its Variants. 2014.

[2] Fortunato Santo. Community detection in graphs. 2009.

[3]Wondracek G , Holz T , Kirda E , et al. A Practical Attack to De-anonymize Social Network Users. 2010.

[4] Remy, Rym, Matthieu. Tracking bitcoin users activity using community detection on a network of weak signals. 2017.