PRIVACY-PRESERVING FUNCTION COMPUTATION BY EXPLOITATION OF FRIENDSHIPS IN SOCIAL NETWORKS

Farid M. Naini, Jayakrishnan Unnikrishnan, Patrick Thiran, Martin Vetterli

School of Computer and Communication Sciences, EPFL, Switzerland
{farid.movahedinaini, jay.unnikrishnan, patrick.thiran, martin.vetterli}@epfl.ch

ABSTRACT

We study the problem of privacy-preserving computation of functions of data that belong to users in a social network under the assumption that users are willing to share their private data with trusted friends in the network. We demonstrate that such trust relationships can be exploited to significantly improve the tradeoff between the privacy of users' data and the accuracy of the computation. Under a one-hop trust model we design an algorithm for partitioning the users into circles of trust and develop a differentially private scheme for computing the global function using results of local computations within each circle. We quantify the improvement in the privacy-accuracy tradeoff of our scheme with respect to other mechanisms that do not exploit inter-user trust. We verify the efficiency of our algorithm by implementing it on social networks with up to one million nodes. Applications of our method include surveys, elections, and recommendation systems.

1. INTRODUCTION

Several applications such as surveys, elections, and auctions require the computation of functions of private data belonging to multiple users. As an example, consider the network of Netflix users. The private data are the individual users' movie ratings and the global function is the average movie ratings across all users. The challenge in such applications is to perform the computation *accurately* while preserving the *privacy* of the users' data. A vast amount of literature on this topic investigates strategies that can be adopted by the users and/or the service provider (also called server) to enhance the privacy of the users' data. Most of the known non-cryptographic solutions to this problem can be viewed as belonging to one of the following two extreme regimes.

The first regime (Regime I) is when every user trusts only herself, not the server nor other users, and she is responsible for protecting her own privacy. In other words, the "circle of trust" of a user comprises only herself. She can protect her privacy by, for example, adding some random noise to her private information before sending it to the server. Clearly the addition of noise leads to a reduction in the accuracy of the computed global function, which is known as the privacy-accuracy tradeoff (also referred to as the privacy-utility tradeoff). In the second regime (Regime II), every user trusts herself and the server but not any of the other users. In other words, the circle of trust of a user comprises herself and the server. In this regime, each user is willing to send her exact private information to the server, and expects the server to protect the privacy of their data.

Both these regimes have some inherent drawbacks. In Regime I, typically the accuracy of the computation has to be compromised significantly in order to obtain sufficient privacy. In Regime II, the users need to trust the server completely. If, for instance, the server discloses the users' information to a third-party, then their privacy may be compromised even if the released data were anonymized [1].

In this context we propose an alternative to privacy enhancement methods proposed for these two regimes. In practice, a user often has trusted "friends" with whom she is willing to share her private information and whom she trusts to perform computations accurately and protect the privacy of her information. We consider such a regime in which the circle of trust of a user consists of herself and her friends, but not the server. This regime is hence a middle ground between the two extreme regimes I and II. The trust relationships between users are represented in the form of connections in a social network. The key idea that we introduce in this paper is that the knowledge of the social network can be intelligently exploited to design function-computation schemes that perform better in terms of privacy-accuracy tradeoff, when compared to Regime I. We first partition the users into circles of trust based on the underlying social network. The users within each circle perform local computations and the results of these computations are then transmitted to the server in a privacy-preserving manner, where the final global function is computed. As the individual user's data is *hidden* within the local computations, this approach yields an additional layer of protection compared to schemes in Regime I.

The existing literature on privacy-preserving function computation can be divided into two broad categories. One category is *perturbation* methods, where users' data are perturbed to protect their privacy [2], for example by addition of random noise [3, 4]. For this category of methods, the circle of trust of a user comprises only herself if she applies the perturbation technique herself, and herself and the server if she trusts the server to perform the perturbation technique. Another category is *cryptographic* methods, where users' data are encrypted to ensure their privacy [5–10]. Some of these techniques belong to the class of Secure Multiparty Computation (SMC) protocols, where users can compute functions of their private data in a distributed way such that every user learns only the value of the output function and nothing more about other users' private data. In this case, the circle of trust of each user consists of only herself. These methods usually need extensive computational power [11].

In this paper we use differential privacy (DP) for quantifying a user's privacy with respect to the server and other entities outside of her circle of trust. Differential privacy is a well accepted notion of privacy for database privacy [12]. It models users' data as deterministic, and gives a strong guarantee that the perturbation of a single user's data will not significantly affect the output of the computation. Differential privacy quantifies a worst-case guarantee with respect to an adversary who may have access to auxiliary information about the users' data. In this work, we develop a perturbation-based function-computation scheme in which all users are guaranteed the same level

This work was supported by ERC Advanced Investigators grant no. 247006 and by Nokia Research Center grant 'Accidental sampling'.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

of differential privacy with respect to the server. We use the mean-squared-error (MSE) in estimating the global function of interest as the accuracy measure. We compare the performance of our scheme with those under Regime I that do not exploit inter-user trust, and show that our scheme performs better in terms of privacy-accuracy tradeoff. Other privacy measures used in the literature include confidence intervals [3], mutual information [13, 14], priori and posteriori knowledge [15, 16], game theory [17], and cryptographic notions of confidentiality [18, 19]. For a more comprehensive review on the literature we refer the reader to the surveys of [20, 21].

The rest of the paper is organized as follows. We introduce the problem and motivate the proposed approach in Sec. 2. We compute the privacy-accuracy tradeoff of our scheme in Sec. 3. We describe our star cover algorithm in Sec. 4. In Sec. 5 we discuss experimental evaluation of our method and conclude in Sec. 6.

2. PROBLEM DESCRIPTION

2.1. Model

We model the friendship (trust) relationships among all the users in a social network by a "friendship" graph $G = (V, E)$, which we assume is a simple undirected connected graph with vertex set V, consisting of $N \geq 3$ nodes $v_1, v_2, \dots, v_N$ representing the users and edge set E. An example is the graph shown in Figure 1(a). An edge $e_{ij} \in E$ exists if and only if users v_i and v_j are friends, indicating that they trust each other and are willing to receive and perform computations on information from each other (i.e., we assume a *one-hop trust* model). We use the terms node and user interchangeably. Every user v_i has some private information denoted by X_i that takes values in some bounded interval $\mathcal{S} \subset \Re$ whose length is denoted by $|\mathcal{S}|$, where $\Re$ is the set of real numbers. We assume X_i's are deterministic. The objective is to compute some global function $f(X_1, X_2, \dots, X_N)$ of the private information of all users. We assume that function f belongs to the class of *divisible* functions [22], i.e., it admits a decomposition of the form

$$f(X_1, X_2, \dots, X_N) = g\left(h_1(Z_1), h_2(Z_2), \dots, h_M(Z_M)\right) \quad (1)$$

for all partitions $\{Z_j\}_{1 \leq j \leq M}$ of the set of variables $\{X_i\}_{1 \leq i \leq N}$. Examples of functions that have this property include sum, product, arithmetic mean, histogram, minimum and maximum functions. In this paper, we restrict our privacy analysis to the sum function

$$f(X_1, X_2 \dots, X_N) = \sum_{i=1}^{N} X_i = X_{sum} \quad (2)$$

for real-valued inputs, which is the function of interest in applications like census surveys and recommendation systems, with the understanding that the results presented here can be generalized to other divisible functions and other ranges of data values.

2.2. Proposed Approach

In our approach the server uses its knowledge of the topology of G to partition users into clusters. Since users are willing to share information only with immediate neighbors under the one-hop trust model, each cluster must have one user who is a neighbor of every user in the cluster. Equivalently, each cluster must be a star subgraph of G. A star is a tree with maximum diameter 2 [23]. Given a friendship graph G, we first identify a spanning star forest of the graph G, i.e., a spanning subgraph of G whose connected components are stars. We denote these stars by $S^1, S^2, \dots, S^r$, where r is the number of connected components in the spanning subgraph. We call $\mathcal{C} = \{S^1, S^2, \dots, S^r\}$ a *star cover* of graph G. Note that the stars cover the nodes of the graph, but not necessarily the edges. In general star covers are not unique. The exact criterion and algorithm we use for choosing the appropriate star cover are described in Sec. 4. We denote by c_j the center node of star S^j, and by k_j the number of vertices in S^j. The center nodes $c_1, c_2, \dots, c_r$, also called star centers, form a *dominating set* for G, i.e., every vertex in G is at most one hop away from one of the star centers [23]. We use $s(i)$ to denote the index of the star to which user i is assigned. Figure 1(b) shows an example star cover $\mathcal{C} = \{S^1, S^2\}$ of the graph in Figure 1(a) with $r = 2$ stars. Star S^1 consists of $k_1 = 3$ nodes v_5, v_6, v_7, where v_5 is the center node ($c_1 = v_5$), and star S^2 consists of $k_2 = 4$ nodes v_1, v_2, v_3, v_4, where v_3 is the center node ($c_2 = v_3$).

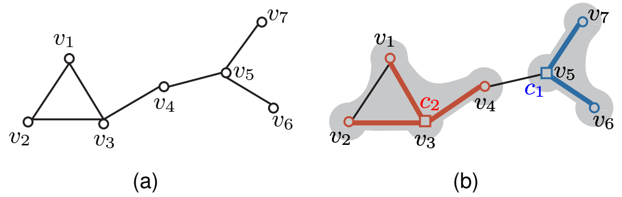

Fig. 1. (a) An example of a friendship graph G with 7 nodes. (b) A spanning star forest of G in (a). Square shaped nodes and thick edges represent centers and edges of the stars. Each star represents a circle of trust (enclosed by a gray area in the figure). Each star center collects the information from neighboring nodes in the star and sends the sum of the collected information to the server.

By partitioning the graph into disjoint star-shaped clusters we effectively partition users into disjoint circles of trust where all users in each circle (i.e., star) trusts the user represented by the star center. In our scheme, the center node c_j of each star S^j collects the private information of all the nodes, computes the sum

$$Z_j = \sum_{v_i \in S^j} X_i$$

of the collected information and sends a perturbed version ($\widetilde{Z}_j$) of it to the server. The server computes sums up the local values it receives from the star centers to obtain

$$\widehat{X}_{sum} = \sum_{j=1}^{r} \widetilde{Z}_j,$$

which is an unbiased estimate for X_{sum} in Eq. (2) when the added perturbations are zero-mean. In our approach, we assume that the server knows the friendship graph topology and that it is the entity responsible for performing the star cover.

3. PRIVACY-ACCURACY TRADEOFF

In this section we quantify the privacy guarantee and accuracy obtained using our scheme, and compare them with schemes under Regime I that do not exploit inter-user trust.

3.1. Differential Privacy (DP) Metric

We propose an algorithm for computing the sum of users' data that guarantees ϵ-differential privacy (ϵ-DP) to all the users with respect to the server. Let $\boldsymbol{X} = [X_1, X_2, \dots, X_N]$ denote the vector of users' private information. Denote by $\boldsymbol{\mathcal{T}}(\boldsymbol{X})$ the set of *all information available at the server.* Vector $\boldsymbol{\mathcal{T}}(\boldsymbol{X})$ is essentially a vector of size r that contains all the uploaded values by the star centers. Our proposed algorithm is ϵ-DP (i.e., all users enjoy ϵ-DP) if

$$\mathsf{P}\left[\boldsymbol{\mathcal{T}}(\boldsymbol{X})=\boldsymbol{t}\right] \leq \exp(\epsilon) \times \mathsf{P}\left[\boldsymbol{\mathcal{T}}(\boldsymbol{X}^{'})=\boldsymbol{t}\right] \tag{3}$$

for all pairs $\boldsymbol{X}, \boldsymbol{X}^{'}$ which differ in only one entry, and for all $\boldsymbol{t} \in \Re^r$, where by abuse of notation we use $\mathsf{P}\left[\boldsymbol{\mathcal{T}}(\boldsymbol{X})=\boldsymbol{t}\right]$ to denote the probability density function of $\boldsymbol{\mathcal{T}}(\boldsymbol{X})$ computed at $\boldsymbol{t}$ [24].

Differential privacy guarantee enjoyed by a user gets stronger as ϵ decreases to 0. Differential privacy quantifies a worst-case guarantee with respect to an adversary who may have access to auxiliary information about the users' data. In particular, even if all users outside the circle of trust of a user decide to collude and report their data to an adversary, the user still has ϵ-DP. Therefore, our approach is robust to malicious nodes outside a user's circle of trust. Furthermore, DP is composable. That is, joint computation of functions $f_1, f_2, \ldots, f_q$, each with $\epsilon_1, \epsilon_2, \ldots, \epsilon_q$-DP guarantee, respectively, yields $\left(\sum_{l=1}^{q} \epsilon_l\right)$-DP guarantee [25].

A popular mechanism to guarantee differential privacy [12, 26] in function computation is via the Laplacian mechanism. In this approach, the value of the computation is perturbed using additive Laplacian noise. In order to guarantee ϵ-DP for computing a real-valued function f via the Laplacian mechanism, the result of the computation $f(\boldsymbol{X})$ is perturbed by adding to it a noise term that follows a Laplacian distribution with mean 0 and variance $2\left(\Delta(f)/\epsilon\right)^2$, where $\Delta(f)$ is the sensitivity of the function f defined as

$$\Delta(f) = \max_{\boldsymbol{X}, \boldsymbol{X}^{'}} \left| f(\boldsymbol{X}) - f(\boldsymbol{X}^{'}) \right|, \tag{4}$$

where the maximum is taken over all pairs $\boldsymbol{X}, \boldsymbol{X}^{'}$ which differ in only one entry. The Laplacian mechanism is known to be optimal for high privacy scenario ($\epsilon \rightarrow 0$) [27]. For the purpose of analysis, we restrict ourselves to the Laplacian mechanism.

3.2. Quantifying Accuracy

To guarantee ϵ-DP, each star center adopts a Laplacian mechanism while reporting the value of the local computation to the server. In the case of sum function, it is easy to see that it suffices that each star center c_j uploads

$$\widetilde{Z}_j = Z_j + n_j, \tag{5}$$

to the server, where n_j is zero-mean Laplacian noise with variance $\sigma_n^2 = 2|\mathcal{S}|^2/\epsilon^2$.

At the server, an estimate $\widehat{X}_{sum}$ of X_{sum} is computed by $\widehat{X}_{sum} = \sum_{j=1}^{r} \widetilde{Z}_j$. There are r independent noise terms added to X_{sum}, hence the MSE in estimating X_{sum} using $\widehat{X}_{sum}$, which is defined as our accuracy measure, is

$$\text{MSE}(\widehat{X}_{sum}) = 2r|\mathcal{S}|^2/\epsilon^2. \tag{6}$$

Note that a *small* MSE indicates that X_{sum} can be accurately estimated, i.e., *high* accuracy. It is immediately seen that if users demand more privacy (smaller ϵ), the accuracy in computing X_{sum} degrades, which indicates the privacy-accuracy tradeoff of our scheme. In addition, for a fixed ϵ, the accuracy improves as r decreases. In other words, a star cover with fewer star components, performs better in the privacy-accuracy tradeoff compared to a star cover with more star components.

For comparison, now consider a perturbation-based scheme under Regime I that does not exploit inter-user trust. In such a scheme, as every user takes care of her privacy, in order to guarantee ϵ-DP to her data, she has to communicate X_i to the server in an ϵ-DP way. The obtained accuracy is thus

$$\text{MSE}(\widehat{X}_{sum}) = 2N|\mathcal{S}|^2/\epsilon^2, \tag{7}$$

which is a factor N/r *worse* than that obtained under our scheme. The Relative Accuracy Gain (RAG) under the Laplacian mechanism is thus

$$\text{RAG} = \frac{N}{r}. \tag{8}$$

Our scheme thus performs better in the privacy-accuracy tradeoff compared to the perturbation-based scheme in Regime I that does not exploit inter-user trust.

3.3. Extensions of the Method

Our approach can be easily adapted for other divisible functions satisfying (1). For example, in the case of the maximum (resp., minimum) function, each star center uploads to the server the maximum (resp., minimum) of the data within the star in an ϵ-DP way. Suppose that in Eq. (1) $M = r$ and $\{Z_j\}_{1 \leq j \leq r}$ represents the partition induced by the star cover, then the amount of noise added by the star center node c_j to the uploaded data in Eq. (5) is determined from the sensitivity of h_j. Furthermore, if one or more such functions are computed, we can quantify the privacy guarantee of our scheme from the composability property of DP.

The privacy guarantee of our scheme can be enhanced when combined with other privacy-preserving mechanisms. For instance, privacy of the proposed scheme with respect to the server can be boosted by using an SMC protocol [20, 21] for the communication between the star centers and the server. In addition, an extra layer of protection with respect to other users in the cluster can be added if the users in each star adopt an SMC scheme when reporting the data to their star-center.

4. STAR COVERING ALGORITHM

For a general graph G, the problem of finding a star cover as discussed in Sec. 2.2 does not admit a unique solution. A natural choice for the star cover is one that minimizes the number of star components r. The following proposition is immediate from Sec. 3.

Proposition 1. *Consider a star cover $\mathcal{C}^*$ that has the minimum number of star components r among all the possible star covers for a given graph G. Equivalently, $\mathcal{C}^*$ is a star cover whose centers form a* minimum dominating set *(MDS) for G. Then, the MSE in estimating the sum of users' private information is minimized (maximum accuracy) where ϵ-DP is guaranteed to all the users.* □

An MDS of a graph is a dominating set with the minimum number of vertices. Although finding an MDS for a graph is NP-complete, it can be approximated. In our experiments, we use the greedy approximation algorithm [28] initialized with a *warm start* obtained from the solution to a linear program (LP) approximation [29] of the MDS problem. The solution to the LP provides also a lower-bound on the MDS size. Note that $\mathcal{C}^*$ in Proposition 1 may not be unique. Thus even if we identify an approximate MDS for the friendship graph, we still have the task of assigning all the users in the network to the star centers in order to have a complete description of the star cover. We assign the remaining nodes such that the workload of the star centers is balanced. The computation and communication load for each star center node is proportional to the number of users in its star (the star size). Thus the maximum workload among all users can be minimized by choosing the assignment that minimizes the maximum size of all stars, which is called the *rooted* minmax star cover problem

Our overall algorithm for finding a star cover is as follows. In the first stage, we find a set of r star centers (also referred to as star roots) that *approximate* an MDS. In the second stage, we run our rooted minmax star cover algorithm described in Algorithm 1. In the algorithm, Δ_{max} denotes the maximum degree of the roots in G, and G_B denotes the bipartite graph consisting of the set of r roots, the set of $N-r$ remaining nodes and the edges between the two sets. For integer-valued vector $\mathsf{REP} = [\mathsf{REP}(1), \mathsf{REP}(2), \ldots, \mathsf{REP}(r)]$, we use G_B^{REP} to denote the bipartite graph obtained from G_B by *replicating* every root i together with all its incident edges $\mathsf{REP}(i)$ times. Our rooted minmax star cover algorithm is based on replicating the star centers in G_B^{REP} and performing a maximum cardinality matching; user i is assigned to star center c_j iff node i is matched to one of the replicates of root c_j. The algorithm keeps decreasing the number of root replicates in a greedy manner such that the maximum number of times a star center is replicated is minimized. The algorithm terminates as soon as a replicated root is not matched to any user. We have the following theorem. We do not include a proof due to space constraints.

Theorem 1. *Given r star centers (roots) such that the minimum degree of r_i's is at least 2, the minmax star cover algorithm in Algorithm 1 outputs a star cover with the given roots such that the size $\overline{k}^*$ of the largest star is minimized.* □

```
$\mathsf{REP}(s) \leftarrow \Delta_{max}$ for $s = 1, 2, \ldots, r$;
for $k^* = \Delta_{max}$ to $1$ do
    while $\exists c_i$ s.t. $\mathsf{REP}(i) > k^*$ do
        $\mathsf{REP}(i) \leftarrow \mathsf{REP}(i) - 1$;
        $M \leftarrow$ A maximum matching on $G_B^{\mathsf{REP}}$;
        if $|M| < N - r$ then
            $\mathsf{REP}(i) \leftarrow \mathsf{REP}(i) + 1$;
            $\overline{k}^* \leftarrow k^* + 1$;
            $M \leftarrow$ A maximum matching on $G_B^{\mathsf{REP}}$;
            foreach matched user in $M$ do
                Assign user to star with center $c_j$ if she is
                matched to a replicate of $c_j$;
            end
            Exit the algorithm;
        end
    end
end
```

Algorithm 1: Our proposed rooted minmax star cover algorithm.

In the next section we apply the proposed star covering algorithm on friendship graphs obtained from real datasets.

5. EXPERIMENTS AND DISCUSSION

We apply our star covering scheme on two datasets. Dataset A is a collection of several users' *ego networks*, i.e., the nodes in the graph are a few randomly chosen users and their friends, collected from Google+ [30]. Dataset B consisting of friendship relationships of Pokec, a popular social network [31]. For each dataset, we built graph G by drawing an edge between any pair of users who are friends with each other (bidirectional friendships) and discarding all isolated nodes. We then found an approximate MDS (AMDS) for each graph and ran our rooted minmax star cover algorithm. In Table 1 we present statistics of the above graphs including number of nodes N, average degree δ_{avr} and average clustering coefficient C [32], the AMDS size r, and the minmax workload $\overline{k}^*$. The obtained AMDS sizes are within 0.7% of the LP lower-bound. As the graph for dataset A is a collection of a few users' ego networks, the graph is inherently well connected. Thus, the graph has a small MDS.

G	Graph Statistics				$\overline{k}^*$	RAG
	N	δ_{avr}	C	r		
A	95897	30.4	0.40	153	3278	626.7
B	1198274	13.9	0.11	209360	223	5.72

Table 1. Statistics of the friendship graphs obtained from the datasets; resulting minmax workload $\overline{k}^*$; and the Relative Accuracy Gain (RAG) (refer to Eq. (8)) of our scheme relative to perturbation-based schemes under Regime I that do not exploit inter-user trust.

When all users are guaranteed ϵ-DP, the Relative Accuracy Gain (RAG) (refer to Eq. (8)) in estimating X_{sum} compared to perturbation-based schemes under Regime I is 626.7 and 5.72 for datasets A and B, respectively. Hence, our proposed scheme performs much better, especially for dataset A, compared to perturbation-based schemes under Regime I that do not exploit inter-user trust. The benefits of our scheme are enjoyed especially when the size r of the AMDS is small compared to the total number of users N. This happens for example when the graph is well-connected such as that for dataset A.

6. CONCLUSION AND FUTURE WORK

We studied the problem of privacy-preserving computation of functions of data belonging to users in a social network under the assumption that users are willing to trust their friends with their data. Our approach is based on partitioning the friendship graph into disjoint circles of trust, and performing local computations within each circle. In a setting where all users are guaranteed ϵ-differential privacy, the distortion added to the computed global function under our scheme is much lower compared to that under a scheme that does not exploit inter-user trust. In addition, our algorithm for partitioning the friendship graph ensures that the workload on star centers are balanced. From the experimental evaluation of our algorithms on real social networks, we observed that the algorithm provides good privacy-accuracy tradeoff when the graph is well-connected. As we discussed it is also possible to extend our scheme to more general functions. We are currently exploring applications of this scheme in social computing applications like crowd-sourcing and participatory sensing.

7. ACKNOWLEDGMENTS

We thank Ola Svensson for helpful discussions on the star covering algorithm and Anand Sarwate for helpful discussions on the Laplacian mechanism.

8. REFERENCES

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 111–125.

[2] Y. Sang, H. Shen, and H. Tian, "Effective reconstruction of data perturbed by random projections," *Computers, IEEE Transactions on*, vol. 61, no. 1, pp. 101–117, 2012.

[3] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM Sigmod Record*, vol. 29, no. 2, pp. 439–450, 2000.

[4] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 37–48.

[5] S. Zhong, Z. Yang, and T. Chen, "k-anonymous data collection," *Information sciences*, vol. 179, no. 17, pp. 2948–2963, 2009.

[6] C. Aguilar-Melchor, S. Fau, C. Fontaine, G. Gogniat, and R. Sirdey, "Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain," *Signal Processing Magazine, IEEE*, vol. 30, no. 2, pp. 108–117, 2013.

[7] J. Bringer, H. Chabanne, and A. Patey, "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *Signal Processing Magazine, IEEE*, vol. 30, no. 2, pp. 42–52, 2013.

[8] C. Orlandi, "Is multiparty computation any good in practice?," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 5848–5851.

[9] P. Smaragdis and M. V. S. Shashanka, "A framework for secure speech recognition," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 2007, vol. 4, pp. IV–969–IV–972.

[10] R. Lazzeretti and M. Barni, "Private computing with garbled circuits [applications corner]," *Signal Processing Magazine, IEEE*, vol. 30, no. 2, pp. 123–127, 2013.

[11] Y. Sang and H. Shen, "Efficient and secure protocols for privacy-preserving set operations," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 1, pp. 9, 2009.

[12] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, pp. 1–19. Springer, 2008.

[13] D. Varodayan and A. Khisti, "Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 1932–1935.

[14] L. Sankar, W. Trappe, K. Ramchandran, H. V. Poor, and M. Debbah, "The role of signal processing in meeting privacy challenges: An overview," *Signal Processing Magazine, IEEE*, vol. 30, no. 5, pp. 95–106, 2013.

[15] J. Yao and P. Venkitasubramaniam, "Maximizing privacy in variable bit rate coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 2959–2963.

[16] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2003, pp. 202–210.

[17] N. Zhang, W. Zhao, and J. Chen, "Performance measurements for privacy preserving data mining," in *advances in knowledge discovery and data mining*, pp. 43–49. Springer, 2005.

[18] B. Gilburd, A. Schuster, and R. Wolff, "k-ttp: a new privacy model for large-scale distributed environments," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 563–568.

[19] Z. Erkin, T. Veugen, and R. L. Lagendijk, "Generating private recommendations in a social trust network," in *Computational Aspects of Social Networks (CASoN), 2011 International Conference on*. IEEE, 2011, pp. 82–87.

[20] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (CSUR)*, vol. 42, no. 4, pp. 14, 2010.

[21] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 5, 2009.

[22] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 4, pp. 755–764, 2005.

[23] D. West, *Introduction to graph theory*, vol. 2, Prentice hall Englewood Cliffs, 2001.

[24] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, pp. 265–284. Springer, 2006.

[25] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 493–502.

[26] C. Task and C. Clifton, "A guide to differential privacy theory in social network analysis," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining*. IEEE Computer Society, 2012, pp. 411–417.

[27] Q. Geng and P. Viswanath, "The optimal mechanism in differential privacy," *arXiv preprint arXiv:1212.1186*, 2012.

[28] D. J. Marchette, *Random Graphs for Statistical Pattern Recognition*, Wiley, 2005.

[29] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1999.

[30] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 548–556.

[31] L. Takac and M. Zabovsky., "Data analysis in public social networks," in *International Scientific Conference AND International Workshop Present Day Trends of Innovations*, Poland, 2012.

[32] D. J. Watts and S. H Strogatz, "Collective dynamics of small-worldnetworks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.