Aalborg Universitet

AALBORG UNIVERSITY

Privacy-Preserving Distributed Average Consensus based on Additive Secret Sharing

Li, Qiongxiu; Cascudo, Ignacio; Christensen, Mads Græsbøll

Published in:
EUSIPCO 2019 - 27th European Signal Processing Conference

DOI (link to publication from Publisher):
10.23919/EUSIPCO.2019.8902577

Publication date:
2019

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Li, Q., Cascudo, I., & Christensen, M. G. (2019). Privacy-Preserving Distributed Average Consensus based on Additive Secret Sharing. In *EUSIPCO 2019 - 27th European Signal Processing Conference* IEEE Signal Processing Society. https://doi.org/10.23919/EUSIPCO.2019.8902577

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: December 05, 2025

Privacy-Preserving Distributed Average Consensus based on Additive Secret Sharing

Qiongxiu Li[1], Ignacio Cascudo[2], Mads Græsbøll Christensen[1]
[1]Audio Analysis Lab, CREATE, Aalborg University, Aalborg, Denmark
[2]Department of Mathematical Science, Aalborg University, Aalborg, Denmark

***Abstract*—One major concern of distributed computation in networks is the privacy of the individual nodes. To address this privacy issue in the context of the distributed average consensus problem, we propose a general, yet simple solution that achieves privacy using additive secret sharing, a tool from secure multiparty computation. This method enables each node to reach the consensus accurately and obtains perfect security at the same time. Unlike differential privacy based approaches, there is no trade-off between privacy and accuracy. Moreover, the proposed method is computationally simple compared to other techniques in secure multiparty computation, and it is able to achieve perfect security of any honest node as long as it has one honest neighbour under the honest-but-curious model, without any trusted third party.**

***Index Terms*—Distributed average consensus, additive secret sharing, privacy preserving, secure multiparty computation**

I. Introduction

The consensus problem has received a lot of attention from researchers over the past decades since it has many practical uses, such as distributed data fusion [1] and group coordination [2]. To solve the average consensus problem in arbitrary random connected distributed networks (e.g., in wireless sensor networks), many distributed averaging algorithms have been proposed, such as basic average consensus algorithms [3], gossip algorithms [4], [5], ADMM [6] and PDMM [7] algorithms based on convex optimization. These iterative approaches require to exchange information among participants to compute the average result. However, the information exchange is a cause for concerns with respect to the privacy of the data, as private information may be revealed.

To achieve privacy-preserving solutions in the average consensus problem, two categories of algorithms have been proposed. The first type of algorithms [8]–[12] implements average consensus by modifying the basic average consensus algorithm [3] based on the concept of differential privacy [13]. If there are two databases that differ only in one single element, it is easy to get the information of this element by comparing the query results of two databases. Differential privacy aims at protecting the privacy of this single element by introducing randomness in query results. The underlying idea is to maintain a balance between the individual privacy and output accuracy by inserting noise to obfuscate the function output in a random manner. Many algorithms [8]–[12] applied this idea to achieve privacy-preserving average consensus with a careful zero-sum noise insertion process. A detailed analysis of the trade-off between maximum information disclosure and estimation accuracy is performed in [14]. However, as proven in [12], exact accuracy and differential privacy cannot be obtained at the same time. Thus, [15] refers to differential privacy methods as consensus-perturbing algorithms and proposed a new consensus-preserving algorithm that, with the help of a trusted third party, assigns to each node a single noise value the sum of which equals zero. Unfortunately, the trusted third party assumption is not practical in many real-world applications. Another type of algorithms [16]–[19] applies garbled circuits (GC) [20], [21] and homomorphic encryption (HE) [22], [23] techniques, as known from secure multiparty computation [24], to general gossip algorithms [4] to preserve privacy. Secure multiparty computation allows all nodes in a network to jointly compute a function and keep their inputs private. Two GC based gossip algorithms were proposed in [16] to iteratively compare the state values of two nodes and update the state values with a step-size while keeping each state value secret. However, the computational complexity is big and only asymptotic consensus is achieved. The step-size parameter also requires global information beforehand. The HE technique was applied in [18], [19] to compute the consensus in the encrypted domain. The initial state value of each node is kept private because only encrypted values are accessed by other nodes. Unfortunately, the computational complexity of the HE technique is big and a trusted third party is also required.

In this paper, we propose a general, yet simple algorithm to solve the privacy-preserving distributed average consensus problem using the principle of additive secret sharing. Note that additive secret sharing has been applied in various applications such as smart grids [25] to address privacy concerns under a strong assumption of network topology (e.g., fully connected). This differs significantly from the proposed algorithm, since we here assume a more practical and general network topology (i.e., arbitrarily connected). In a decentralized network, the average consensus is usually computed by iterative distributed averaging algorithms such as [3]–[7]. Thus, the question of how to achieve the privacy concern during all iterations is the main challenge.

The proposed approach is lightweight compared to the above mentioned HE and GC approaches in [16]–[19], as only additions are involved. The underlying idea of additive secret sharing is to replace each initial state value with another obfuscated value by subtracting and adding random numbers. Unlike the differential privacy approaches [8]–[12], there is no

trade-off between privacy and estimated accuracy. The main properties of the proposed approach can be summarized as follows: 1) the proposed approach achieves perfect security and exact accuracy at the same time; 2) it is computationally simple; 3) individual privacy is guaranteed as long as it has one honest neighbour under the honest-but-curious model without any trusted third party; 4) it is convenient since only an additive randomization step is needed; and 5) it is very general since it can be applied in any distributed averaging algorithm.

II. Preliminaries and Problem Setup

A. Privacy-preserving distributed average consensus problem

A distributed system composed of a set of nodes can be modelled as an undirected connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The node set of the graph is denoted as $\mathcal{N} = \{1, 2, ..., n\}$ and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the communication links between nodes. The communication of two nodes is enabled if there is one edge connecting two nodes, i.e., $(i, j) \in \mathcal{E}$, and $n_i = \{j | (i, j) \in \mathcal{E}, j \neq i\}$ denotes the neighbours of node i. The initial state value held by node i is denoted as a_i, and the initial state values in the network can be written as a vector $\mathbf{a} = [a_1, a_2, ..., a_n]^T$. The main goal is to address the following two challenges at the same time:

1) Compute the average result of the private values
$$a_{\text{ave}} = \frac{1}{n} \sum_{i=1}^{n} a_i, \tag{1}$$
in a distributed network without having any centralized coordinator, an iterative algorithm is usually adopted.
2) Protect the private value a_i of each node throughout the algorithm execution.

B. Privacy concern and adversary model

An important aspect of this work is the definition of privacy. Our goal here is to protect the initial state value a_i of each node, which constitutes the private information, during the execution of the algorithm. The reason is that it may represent an individual's opinion [26] or private information [18], [19].

In this paper, a general honest-but-curious (also called "passive" or "semi-honest") model is considered. It means that all nodes in the network follow the designed protocol, but some of them might be curious about the other nodes' private information. Such curious nodes are said to be passively corrupted, and they can cooperate to share their received information with the aim of inferring other honest nodes' private information, here the initial state values. We assume the worst case situation within this model, where passively corrupted nodes know the following:

- The whole graph topology.
- The initial state values of all passively corrupted nodes.
- The transmitted information over the communication links involving the corrupted nodes.

Thus, the corrupted nodes will know all the information except the information kept by the honest nodes themselves and exchanged between every two honest nodes, as long as it cannot be deduced from the above.

III. Additive secret sharing

A secret sharing scheme is a cryptographic tool that splits a secret into a number of shares, where each node in a group will receive one share. The secret can be reconstructed only if a sufficient number of shares are collected, otherwise no information about the hidden secret will be revealed. General secret sharing schemes usually consist of two parts:

- The secret sharing algorithm takes a secret s as input and some randomness r, and outputs n shares of this secret:
$$F_{\text{S}}(s, r) = (s_1, s_2, ..., s_n). \tag{2}$$
- The secret reconstruction function (which technically is a family of functions, one for each subset of shares that can reconstruct the secret) takes the shares of some subset of the nodes $\{\Lambda_1, \ldots, \Lambda_t\}$ as inputs to reconstruct the secret s:
$$F_{\text{R}}(s_{\Lambda_1}, s_{\Lambda_2}, ..., s_{\Lambda_t}) = s, \tag{3}$$
where s_i denotes the i^{th} share of secret s. If we have a reconstruction function for any set of at least t shares, but no set with less than t shares provides any information about the secret, then the secret sharing scheme can be referred to as (n, t) threshold secret sharing scheme.

One of the simplest secret sharing schemes is the additive secret sharing where $t = n$. This is defined over an algebraic group F, usually given by the integers $\{0, \ldots, p-1\}$ together with the additive operation modulo p. While p is a prime number in many applications (so that the group is also a finite field), this is not required here. The additive secret sharing scheme is defined as follows: choose $n-1$ integers $r_1, ..., r_{n-1}$ in F uniformly at random. Then the output of the function (2) consists of $s_i = r_i$ for $i = 1, \ldots, n-1$ and $s_n = (s - \sum_{i=1}^{n-1} r_i) \bmod p$. Given the full set of n shares, the secret can be reconstructed by
$$s = (\sum_{i=1}^{n} s_i) \bmod p. \tag{4}$$
It is easy to see that the secret s cannot be reconstructed even if only one share is missing, and the secret is, in fact, uniformly distributed over the integers within F even though the knowledge of $n-1$ shares is given. Additive secret sharing has the following property, which enables secure computation of additions: if two secrets $s, s' \in F$ are shared among some set of nodes, then the nodes can reconstruct the sum of the secrets, without needing to reconstruct the individual secrets, as follows: each node i locally add the received shares s_i, s'_i and reveal only this sum of shares $h_i = s_i + s'_i \bmod p$ to the other nodes. Then applying the reconstruction function (4) to these share sums h_i will give the sum of the original secrets $s + s' = (\sum_{i=1}^{n} h_i) \bmod p$, without revealing anything else. This can be extended to summing an arbitrary number of secrets, and it can be turned into a secure computation protocol to compute the sum of the secrets of a set of n nodes in a fully connected network, where every node first sends shares of its secret among the full set of nodes, and the process

described above is used to reconstruct only the sum. This is secure against an arbitrary number of passive corruptions (see [24, Section 1.3.1] for further details).

IV. Proposed algorithm

In this section, the details of the proposed algorithm will be described. The algorithm itself is shown in Algorithm 1, where d_i denotes the total number of elements in n_i and T denotes the maximum iteration number, F is the set of integers modulo p for a large enough number p ($p > \sum_{i=1}^{n} a_i$), c denotes the penalty parameter in PDMM algorithm [7].

Algorithm 1 Proposed algorithm

Additive randomization:

1: Each node $i \in \mathcal{N}$ extract d_i random numbers as shares r_i^k with uniform probability in F.

2: Node i sends shares r_i^k to its neighbours $k \in n_i$ and keep the share r_i as.

$$r_i = (a_i - \sum_{k \in n_i} r_i^k) \bmod p. \tag{5}$$

3: Node i receives shares r_k^i from its neighbours $k \in n_i$.

4: Node i updates a_i as the obfuscated value

$$u_i = (r_i + \sum_{k \in n_i} r_k^i) \bmod p. \tag{6}$$

Distributed averaging (e.g., PDMM):

5: Each node initializes the primal variable x_i^0 and dual variable $\xi_{i|j}^0$ as zeros, $i, j \in \mathcal{N}$.

6: For iteration $t = 1, 2, 3, ..., T$

7: Activate node $i \in \mathcal{N}$ randomly with uniform probability.

8: Node i updates x_i^t and broadcasts to its neighbours

$$x_i^t = \frac{u_i + \sum_{k \in n_i} (c x_k^{t-1} + \xi_{k|i}^{t-1})}{1 + c d_i}. \tag{7}$$

9: After receiving x_i^t updates, all neighbouring nodes $k \in n_i$ update the dual variable as

$$\xi_{i|k}^t = -\xi_{k|i}^{t-1} + c(x_i^t - x_k^{t-1}). \tag{8}$$

10: Repeat until the primal variable x_i^t converges

Average consensus computation

11: Each node obtains the average as

$$x_{\text{ave}} = \frac{1}{n}(x_i^T \times n \mod p). \tag{9}$$

The first stage of the algorithm is additive randomization, where each node uses additive secret sharing for distributing shares of its private value a_i to its neighbours. We remark that the difference between this use of additive secret sharing and the one described at the end of the previous section is the assumption of the graph topology. The scheme described in the previous section assumes a fully connected network where each node sends shares to all other nodes. However, a fully connected graph scales poorly in the number of connections. In this paper, we assume an arbitrarily connected graph, which is much more practical and scalable in real-life applications. Each node only sends shares to its neighbours and an iterative distributed averaging algorithm is used afterwards. The main goal of additive randomization is to address the privacy challenge in Section II-A by replacing the private value a_i of each node with an obfuscated value u_i, which can then be revealed. In Section V-C we show exactly how much information this provides to the corrupted nodes. An important observation is that by construction we have that $\sum_{i=1}^{n} a_i = (\sum_{i=1}^{n} u_i) \bmod p$.

After additive randomization, we take the obfuscated values u_i as inputs to a distributed averaging algorithm [3]–[7] to compute the average, which meets the requirements described in Section II-A. Here we apply the asynchronous PDMM algorithm as it has the fastest convergence speed [7]. After convergence, the primal variable x_i^T for all nodes $i \in \mathcal{N}$ will reach the average of obfuscated values, i.e., $x_i^T = \frac{1}{n}\sum_{i=1}^{n} u_i$.

The last part of the proposed algorithm is to compute the final average result by (9) with an assumption of knowing the total number of nodes n. The average result x_{ave} of the proposed algorithm is identical to a_{ave} since

$$a_{\text{ave}} = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{1}{n}((\sum_{i=1}^{n} u_i) \bmod p) = x_{\text{ave}}. \tag{10}$$

Concerning data representation, we remark that the values in the additive randomization process should be integers within the modular domain $\{0, ...p-1\}$ due to the additive secret sharing. Floating point numbers can be scaled up as integers and negative numbers can be represented using modular additive inverse. Note that integers are not required afterwards because additive secret sharing scheme is no longer applied in the distributed averaging step, which is also why the division operation can be used in (7).

V. Experimental results and analysis

A. Experimental results

Simulations are conducted here to investigate the performance of the proposed approach. A random geometric graph [27] with $n = 100$ nodes is simulated and the connectivity of nodes is enabled if their distance is within a radius $\sqrt{\frac{\log n}{n}}$ to have a connected graph with high probability [27]. Based on the same initial state values over the network and additive randomization procedure, the simulation results are demonstrated in Fig. 1, where the solid blue, green, red lines denote the conventional non-privacy concerned random gossip [4], asynchronous ADMM [6] and PDMM [7] algorithms, respectively, and the related dashed lines represent the proposed secure approaches which add additive randomization before the above mentioned conventional algorithms, and the penalty parameters in both ADMM and PDMM are set as 0.4.

As demonstrated in Fig. 1, we can see that the estimated accuracy of all the proposed secure approaches is identical to conventional non-secure approaches. The convergence rate of the proposed approaches will be slightly slower than the

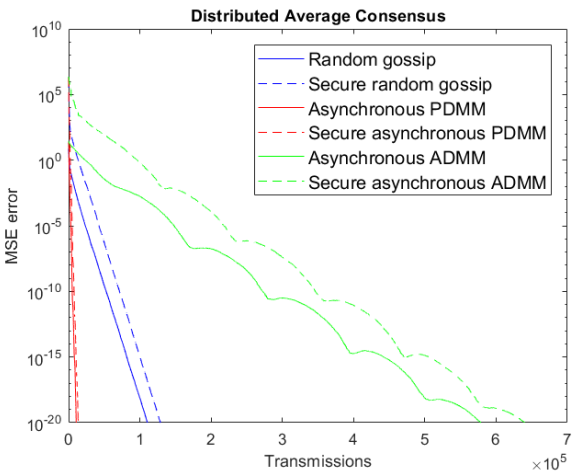

Fig. 1. Experimental results

traditional approaches as the initial mean square error becomes higher after additive randomization. We remark that the extra additive randomization will not affect the convergence speed but only cause higher initial errors.

B. Comparisons

A comparison of the proposed approach with existing methods is shown in Table I, where β denotes the number of bits needed to represent transmitted message [28]. The same passive adversary model is considered in all approaches. We can see that HE and GC based approaches both require expensive computational function as encryption is involved, and the communication bandwidth is also big since the cipher text after encryption usually require much longer bit lengths than plain text, and a trusted third party is also required in HE. Moreover, the proposed approach is able to achieve identical accuracy and perfect security with same communication bandwidth and computational function as differential privacy approaches. Note that the maximum number of corruptions for algorithms without an accuracy trade-off is $n-2$, because the corrupted nodes can always know the initial state value of the only honest node given the knowledge of the exact consensus result and the initial state values of the corrupted $n-1$ nodes. For differential privacy and GC based approaches, the maximum number of corruptions can be $n-1$, as the average result is inexact. Furthermore, the proposed algorithm can protect the privacy of any honest node only if it has at least one honest neighbour, which is not required in the other approaches, e.g., in differential privacy approaches.

C. Security guarantee

In this section, we analyze the security of the proposed algorithm in more detail. The statement we will argue is as follows: Let $\mathcal{C} \subseteq \mathcal{N}$ be the subset of passively corrupted nodes, and let $\mathcal{H} = \mathcal{N} \setminus \mathcal{C}$ be the set of honest nodes. If the subgraph $\mathcal{H}$ is connected, then the only information about the honest nodes' initial state values can be learned by the corrupted nodes is $\sum_{k\in\mathcal{H}} a_k$, but nothing more than that. And we remark that learning this information is logically unavoidable if we have exact accuracy, since this information can always be deduced from the average result and the initial state values of the corrupted nodes:

$$\sum_{k\in\mathcal{H}} a_k = n \times a_{\text{ave}} - \sum_{k\in\mathcal{C}} a_k.$$

This in implies the following: The individual privacy of the honest nodes is protected as long as it has some honest neighbours, even in the case where there are only two honest nodes. If the two honest nodes i, j are neighbours, the corrupted nodes do not learn the individual private value a_i and a_j, but only its sum.

The proof is as follows: adopting a pessimistic view, the information set obtained by $\mathcal{C}$, also known as the information view, after reaching consensus is in the worst case the union $V = V_1 \cup V_2 \cup V_3$ of the information sets

$$\begin{cases} V_1 = \{u_k | k \in \mathcal{N}\}, \\ V_2 = \{a_{\text{ave}}, a_k | k \in \mathcal{C}\}, \\ V_3 = \{r_k^m, r_m^k | k \in \mathcal{C}, m \in d_k\} \cup \{r_k, k \in \mathcal{C}\}. \end{cases}$$

Note that all the obfuscated values u_i in the distributed averaging step are included in V_1 as the primal and dual variables are initialized as zeros in (7), these values u_i can therefore be considered non-private.

Now suppose a "real" instance

$$I = \{a_k, k \in \mathcal{N}\},$$

has produced the above view V with real initial state values and randomness $r_\ell, \ell \in \mathcal{N}$ and $r_\ell^m, (\ell, m) \in \mathcal{E}$.

Let i,j be two honest nodes which are neighbours of each other. We now produce a "fake" instance

$$I' = \{a'_k, k \in \mathcal{N}\},$$

having the view V' with all $a'_k = a_k, k \in \mathcal{N}, k \neq i, j$ and $a'_j = a_j - d$, $a'_i = a_i + d$ for some d. Note that $\sum_{k\in\mathcal{H}} a_k = \sum_{k\in\mathcal{H}} a'_k$ by setting the randomness as $r'_i = r_i + d$, $r_j^{i\,'} = r_j^i - d$, and leave all other random values r_ℓ, r_ℓ^m unchanged.

Thus, the information view V' produced by the fake instance I' will be exactly the same with V produced by the real instance I, which means that the corrupted nodes cannot distinguish the "real" from the "fake". Since $\mathcal{H}$ is connected, we can repeat the argument to modify the initial state values of $\mathcal{H}$ in any way that we want, as long as this modification does not change the sum of the honest initial state values, and still produce the same view for corrupted nodes. The corrupted nodes can only learn the sum of honest nodes' initial state values $\sum_{k\in\mathcal{H}} a_k$, but no other information. Hence, the proposed algorithm is perfectly secure in the sense of secure computation, as it protects all information that is not implied by the average result and corrupted nodes' initial state values.

We remark that if the subgraph of the honest nodes is not connected, then the corrupted nodes can infer the partial sums of the initial state values held by the connected subsets in $\mathcal{H}$, but nothing else beyond that. In an extreme case, if a honest node has only one honest neighbour, then the leaked

TABLE I
PRIVACY-PRESERVING DISTRIBUTED AVERAGE CONSENSUS APPROACHES UNDER ARBITRARY CONNECTED GRAPHS

	Proposed	HE [18], [19]	GC [16]	Differential privacy [8]–[12]
Accuracy	Identical	Identical	Dependent on step size	Degraded with noise
Security	Perfect	Computational	Computational	Differential privacy
Involved function	Linear	Exponential	Exponential	Linear
Trusted Third Party	No	Yes	No	No
Adversary model	Passive	Passive	Passive	Passive
Communication bandwidth per round	$\mathcal{O}(1)$	$\mathcal{O}(\beta)$	$\mathcal{O}(\beta)$	$\mathcal{O}(1)$
Maximum number of corruptions	n-2	n-2	n-1	n-1

information is only the sum of the initial state values held by these two honest nodes, we emphasize that that the privacy of the *individual* node is always protected, which is our goal here. Hence the privacy of individual node is guaranteed as long as it has one honest neighbour.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a general and simple solution to address the privacy concern in distributed average consensus problems with the help of the additive secret sharing scheme. An additive randomization step is applied before distributed averaging to replace the initial state value of each node with a non-private obfuscated value for privacy-preserving. The proposed solution outperforms differential privacy based approaches, as it obtains perfect security and accurate consensus at the same time. Moreover, it is computationally less complex compared to HE and GC based approaches. The proposed algorithm is general and can be used with arbitrary distributed averaging algorithms. Moreover, it does not require any trusted third party, and the privacy of each individual honest node is protected as long as it has one honest neighbour. Future work will focus on how to maintain privacy under more challenging adversary models (i.e., active attacks) where the corrupted nodes may not follow the protocol correctly but deviate from it to interfere with the computation result.

REFERENCES

[1] L. Xiao, S. Boyd, S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *IPSN*, pp. 63–70, 2005.

[2] J. N. Tsitsiklis, "Problems in decentralized decision making and computation.," Tech. Rep., Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1984.

[3] L. Xiao, S. Boyd, "Faster linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65-78, 2004.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[5] A. G. Dimakis, S. Kar, J. M. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[7] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Trans. Signal Process.*, vol. 4, no. 1, pp. 173–187, 2018.

[8] M. Kefayati, M. S. Talebi, B. H. Khalaj, and H. R. Rabiee, "Secure consensus averaging in sensor networks using random offsets," *Proc. of the IEEE Int. Conf. on Telec., and Malaysia Int. Conf. on Commun.*, pp. 556–560, 2007.

[9] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," *ACM workshop Privacy electron. Soc.*, pp. 81–90, 2012.

[10] N. E. Manitara and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," *ECC*, pp. 760–765, 2013.

[11] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Automat Contr.*, vol. 62, no. 2, pp. 753–765, 2017.

[12] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, 2017.

[13] C. Dwork, "Differential privacy," *ICALP*, pp. 1–12, 2006.

[14] J. He, L. Cai, C. Zhao, P. Cheng, and X. Guan, "Privacy-preserving average consensus: privacy analysis and optimal algorithm design," *IEEE Trans. Signal Process.*, 2018.

[15] P. Braca, R. Lazzeretti, S. Marano, and V. Matta, "Learning with privacy in consensus + obfuscation," *IEEE signal process. Lett.*, vol. 23, no. 9, pp. 1174–1178, 2016.

[16] F. Hanzely, J. Konečnỳ, N. Loizou, P. Richtárik, and D. Grishchenko, "Privacy preserving randomized gossip algorithms," *arXiv preprint arXiv:1706.07636*, 2017.

[17] R. Lazzeretti, S. Horn, P. Braca, and P. Willett, "Secure multi-party consensus gossip algorithms," *ICASSP*, pp. 7406–7410, 2014.

[18] R. C. Hendriks, Z. Erkin, and T. Gerkmann, "Privacy preserving distributed beamforming based on homomorphic encryption," *EUSIPCO*, pp. 1–5, 2013.

[19] R. C. Hendriks, Z. Erkin, and T. Gerkmann, "Privacy-preserving distributed speech enhancement forwireless sensor networks by processing in the encrypted domain," *ICASSP*, pp. 7005–7009, 2013.

[20] A. C. Yao, "Protocols for secure computations," *FOCS*, pp. 160–164, 1982.

[21] A. C. Yao, "How to generate and exchange secrets," *FOCS*, pp. 162–167, 1986.

[22] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *EUROCRYPT*, pp. 223–238, 1999.

[23] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," *Advances in Cryptology–CRYPTO*, pp. 643–662, 2012.

[24] R. Cramer, I. B. Damgrd, and J. B. Nielsen, *Secure multiparty computation and secret sharing*, Cambridge University Press, 2015.

[25] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Privacy Enhancing Technologies*. pp. 175–191, 2011.

[26] M. H. DeGroot, "Reaching a consensus," *J. Am. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[27] J. Dall and M. Christensen, "Random geometric graphs," *Physical review E*, vol. 66, no. 1, pp. 016121, 2002.

[28] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Magazine*, vol. 30, no. 1, pp. 82–105, 2013.