A Privacy-Preserving Asynchronous Averaging Algorithm based on Shamir's Secret Sharing

Qiongxiu Li, Mads Græsbøll Christensen
Audio Analysis Lab, CREATE, Aalborg University, Aalborg, Denmark

***Abstract*—Average consensus is widely used in information fusion, and it requires information exchange between a set of nodes to achieve an agreement. Unfortunately, the information exchange may disclose the individual's private information, and this raises serious concerns for individual privacy in some applications. Hence, a privacy-preserving asynchronous averaging algorithm is proposed in this paper to maintain the privacy of each individual using Shamir's secret sharing scheme, as known from secure multiparty computation. The proposed algorithm is based on a lightweight cryptographic technique. It gives identical accuracy solution as the non-privacy concerned algorithm and achieves perfect security in clique-based networks without the use of a trusted third party. In each iteration of the algorithm, each individual's privacy in the selected clique is protected under a passive attack where the adversary controls some of the nodes. Finally, it also achieves robustness of up to one third transmission error.**

***Index Terms*—Distributed average consensus, Shamir's secret sharing, privacy-preserving, active attack, secure multiparty computation**

I. Introduction

Consensus has been intensively investigated over the past decades since it is useful to solve problems in information fusion, especially in distributed systems. Distributed average consensus has been adopted in various applications such as group coordination [1] and dynamic load balancing [2]. There are many approaches to iteratively achieve consensus without centralized coordination: average consensus algorithms [3], general-purpose gossip algorithms [4], [5], methods based on convex optimization such as the ADMM [6] and the PDMM [7] algorithms and graph filter methods [8], [9]. All the algorithms above require information exchange between certain entities. However, this exchange may disclose the individual's privacy. In a distributed network, such as a sensor network, the nodes of the network are interested in reaching an agreement but they may also have concerns about protecting the privacy of their data. For example, a group of individuals may want to achieve a common opinion using a consensus algorithm; at the same time, each individual is unwilling to trust the others by revealing his/her own opinion [10]. This makes privacy-preserving in consensus problem a crucial topic to address.

Two types of methods have been deployed to obtain privacy-preserving solutions in distributed average consensus: differential privacy [11] approaches, which try to maintain the maximum accuracy from statistical database queries while minimizing the chances of identifying its records; and secure multiparty computation [12] approaches, which aim at jointly computing a function over the inputs of a set of nodes while keeping their inputs private. The underlying idea in most existing differential privacy algorithms [13]–[17] is to mask the secret values with zero-sum random noise during the information exchange. This protects privacy without any trusted third party while the average consensus is still achieved by carefully design the noise insertion process. A statistical analysis of maximum disclosure probability and estimation accuracy is performed in [18]. However, Nozari et al. [15] proved that exact average consensus and differential privacy cannot be achieved simultaneously. Differential privacy based algorithms are thus referred to as consensus perturbing approaches [19]. A new consensus preserving approach was proposed in [19] that guarantees an exact average by employing obfuscation via a single noise sample for each node while ensuring that the allocated noise sum to zero. However, the obfuscation noise samples have to be generated by a trusted third party, something that is not always practical.

Other algorithms [20]–[23] obtain secure average consensus based on techniques from secure multiparty computation, such as homomorphic encryption (HE) schemes [24], [25] and the garbled circuit (GC) technique [26], [27]. Homomorphic encryption enables computation on the encrypted data. HE was adopted in [22], [23] to guarantee that each node can only access the encrypted values of other nodes. However, HE requires a high computational complexity for encryption and a trusted third party. Two GC based algorithms were proposed in [20] to securely compare the state value of two nodes. However, these are also computational expensive and requires global information beforehand, and only asymptotic consensus is obtained.

In this paper, Shamir's secret sharing scheme, as known from secure multiparty computation, is adopted in a distributed asynchronous averaging algorithm reminiscent of [3] to solve the problem of privacy-preserving distributed average consensus. The main idea is to divide a secret into a number of shares and distribute a share to each node in the network. The secret can be reconstructed if and only if a sufficient amount of shares are collected, otherwise no information of secret will be disclosed. Compared to differential privacy based approaches [13]–[17], the proposed algorithm is able to achieve perfect security and exact accuracy at the same time. Since only computations on polynomials are involved in Shamir's scheme, it has lower computational complexity compared to encryption approaches such as the HE and GC of [20]–[23] and no trusted third party is required. Moreover, the proposed method

978-9-0827-9703-9/19/$31.00

considers both a general passive attack model and a weak active attack model.

II. Preliminaries and Problem Setup

A. Privacy-preserving distributed average consensus problem

In a distributed system, we assume an undirect connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed by the node set $\mathcal{V} = \{1, 2, ..., n\}$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of undirected edges. Every two nodes can communicate with each other if and only if they are connected neighbours, i.e., $(i, j) \in \mathcal{E}$. The neighbourhood of node i is denoted as $d_i = \{j | (i, j) \in \mathcal{E}, j \neq i\}$. Each node i holds an initial state value $a_i(0)$, which is its private information, and the vector of the initial state values on the network is denoted as $\mathbf{a}(0) = [a_1(0), a_2(0), ..., a_n(0)]^T$. The main goal is to solve the following two challenges at the same time:

1) Compute the average result of the private information
$$a_{ave} = \frac{1}{n} \sum_{i=1}^{n} a_i(0) \tag{1}$$
in a distributed network without having any centralized coordinator, here using an iterative average consensus algorithm.
2) The private information of each node, $a_i(0)$, in the network should be protected during the iterations of the algorithm, hence preserving the privacy of the node.

B. Privacy concern and adversary model

The privacy concern addressed in this paper here pertains to the initial state value held by each node in the network, as it may be sensitive and undesirable for each node if this is revealed to others. The adversary models adopted here include a passive and a very weak kind of active attack model that is interesting from a practical point of view. In the passive attack model, also named honest-but-curious model, each node follows the protocol correctly but so-called passively corrupted nodes try to infer the honest nodes' privacy. A number of passively corrupted nodes may cooperate to increase the chance of inferring the other's initial state value by sharing information. In contrast, in an active attack model the corrupted nodes may not follow the defined protocol and attempt to manipulate the computation result by lying about the exchanged information or refuse to act according to the protocol. In that case, we would need to deal with these active attacks, and this can be done in a number of ways: one possibility is to settle for a solution that detects the errors and aborts the process in time; but a more ambitious possibility would be to find a protocol that does not only detect errors but is also able to correct the errors automatically without aborting, a property usually referred to as robustness. In what follows, we consider a weaker model that is sufficient to achieve robustness towards transmission errors.

III. Shamir's Secret Sharing scheme

In this section, a technique from secure multiparty computation called Shamir's secret sharing is introduced. Before exploring the algorithm in detail, we illustrate the concept of secret sharing with the following example [28]: several scientists are working on a secret project where some confidential documents are in a cabinet locked up with a pass-code. The cabinet can be unlocked if (and only if) half or more of the scientists are present. Shamir's secret sharing scheme [29], first proposed by Shamir in 1979, provides a powerful solution to this problem. The principle of Shamir's secret sharing is Lagrange polynomial interpolation. It is based on the fact that a prior unknown polynomial with degree at most t can be reconstructed if its value at $t+1$ or more points are given, but any number strictly smaller than $t+1$ will give no information about the polynomial in other points.

Shamir's secret sharing is defined as follows. Assume there are n nodes, referred to as $p_1, p_2, \ldots, p_n$. The indices of these nodes are denoted as $\mathcal{N} = \{1, 2, ..., n\}$. Take a finite field F of cardinality more than n, for example we take the field of integers modulo a prime number p with $p > n$. In addition we select an integer $t < n$. In order to share a secret $s \in F$, the dealer (the node who knows the secret) proceeds as follows:

1) **Polynomial construction**: Selects coefficients $\{c_i | i = 1, 2, \ldots, t\}$ uniformly at random in F and constructs the polynomial $f(x) = s + c_1 x + c_2 x^2 + \cdots + c_t x^t \mod p$. Note that the secret is $f(0)$.
2) **Share distribution**: Compute and distribute the secret shares s_i related to p_i as $s_i = f(i) \mod p, i \in \mathcal{N}$ Note that since the dealer is also one of the p_i's, this also includes sending a share to itself. This share will be needed when aggregating the information with the shared secrets from other nodes later on.
3) **Secret reconstruction**: If a set of $t + 1$ nodes, indexed by $\Lambda \subseteq \mathcal{N}$ agree to reconstruct the secret, they can use Lagrange interpolation
$$s = \sum_{i \in \Lambda} r_i s_i. \tag{2}$$
where r_i is the Lagrange basis computed by
$$r_i = \prod_{j \in \Lambda \setminus \{i\}} \frac{-j}{i - j}. \tag{3}$$

This Shamir's secret sharing scheme divides the secret s into several shares $s_i, i \in \mathcal{N}$ and distributes them to n different nodes, and all shares and the secret are evaluations of a polynomial of degree t. The privacy guarantee of Shamir's scheme is based on interpolation properties, implying that a set of t or less shares gives no more information about the secret than what was known a priori. Moreover, Shamir's scheme is also linear: it allows to "add secrets". If two secrets s and s' are shared, possibly by different dealers, among the same network of users by using polynomials f and f', then the nodes can obtain a sharing of $s + s'$ by simply adding their two shares. This works well because $f + f'$ is still of degree $\leq t$ and $f(i) + f'(i) = (f + f')(i) \mod p$.

Finally Shamir's scheme has certain error correction properties that can be used to detect errors, and in some cases, correct them. This allows to correct certain types of active malicious behaviour. More precisely it is a robust secret sharing scheme: if $t < n/3$, then given the set of all n shares, if at most t are erroneous then the Berlekamp-Welch algorithm [30] can output the correct secret. This prevents a set of $t < n/3$ nodes to cheat when reconstructing the secret. Following the above description, the correct polynomial constructed by secret holder is $f(x)$, and the received share set is denoted by $\{(i, s_i), i \in \mathcal{N}\}$. Since there might be some inconsistent shares in the share set, instead of directly constructing $f(x)$ based on Lagrange interpolation, we set to find two other polynomials $e(x)$ with degree t and $q(x)$ with degree $2t$ satisfying the following equality

$$q(x) = e(x)f(x),$$

and where in addition $e(x)$ (referred to as error locator polynomial) satisfies that $e(i) = 0$ whenever $f(i) \neq s_i$. Under the two conditions above, $e(x)$ and $q(x)$ satisfy the following system of linear equations, in which the unknowns are the coefficients of e and q:

$$s_i e(i) = q(i), i \in \mathcal{N}. \tag{4}$$

If we can solve the system and find e and q, then $f(x)$ can be constructed correctly as $f(x) = q(x)/e(x)$. As there are n equalities available with $3t + 1$ coefficients (the coefficient e_t in $e(x) = e_0 + e_1 x + \ldots + e_t x^t$ can be set as 1) in (4), the degree t of the polynomial should be smaller than $n/3$ in order to solve the equation. Thus, the Berlekamp-Welch algorithm allows to correct the secret even in the presence of t invalid shares in secret construction step as long as $t < n/3$. However, it does not prevent malicious behaviour (by even one malicious node) when creating the shares, as this node could create more than $n/3$ errors in the sharing process. This can be detected by the use of verifiable secret sharing [31]. We will not be concerned about this in this paper.

It is important to note that Shamir's secret sharing scheme is only applicable to fully connected graphs due to the fact that each node has to distribute shares to all other nodes. This affects the choice of distributed averaging algorithm and the possibility of graph topology relaxation. We will address these issue in the next section.

IV. Proposed Approach

To approach the challenges of having an algorithm that is both distributed and privacy-preserving, we adopt a distributed asynchronous averaging algorithm based on [3] to compute the average iteratively, and Shamir's secret sharing scheme is then applied in each iteration of this algorithm to guarantee that the privacy of each node is protected. The detailed algorithm is described in Algorithm 1.

As previously mentioned, the application of Shamir's secret sharing scheme requires a fully connected graph, something that was also observed in [32]. However, such graphs are not always practical or scalable since they require a huge number of connections. Therefore, we adopt a distributed asynchronous averaging algorithm to relax the network topology requirement: as shown in step 3 of Algorithm 1, Shamir's secret sharing scheme is applied in a fully connected subset of nodes in each iteration. Thereby, we relax the impractical topology requirement, from a fully connected to a clique-based graph, and a preprocessing step named clique detection is added. The clique C_i of node i should satisfy

$$\begin{cases} C_i \subseteq \{d_i \cup i\}, n_i > 2, \\ \forall j, k \in C_i, j \neq k, (j, k) \in \mathcal{E}, \end{cases} \tag{5}$$

where n_i denotes the total node number in clique C_i. We can see that C_i need not be unique. The requirement $n_i > 2$ is simply due to the fact that one can always infer the other's initial state value with the final addition result if there are only two nodes [33]. The clique-based graph topology is required to guarantee that each node should have at least two neighbour nodes and all these three nodes are interconnected. In practice, the clique based graph is quite normal in distributed system (e.g., in wireless sensor networks) since the connectivity between certain nodes is typically enabled for nodes within a fixed distance of each other.

Algorithm 1 Proposed approach

Clique selection:

1: For all the nodes $i \in \mathcal{V}$ in the whole network, find all possible cliques C_i satisfy (5).

Distributed asynchronous averaging [3]:

2: Randomly activate one node i with uniform probability.
3: Node i choose one clique C_i and set the polynomial degree t based on adversary model, compute the addition result $y(k) = \sum_{j \in C_i} a_j(k)$ securely in selected clique based on Algorithm 2.
4: Update the node values as $a_j(k+1) = \frac{y(k)}{n_i}, j \in C_i$.
5: Repeat step 2-3 till convergence.
6: End

Algorithm 2 describes a solution to securely compute addition in the selected clique C_i based on the linearity of Shamir's secret sharing. The attack model is defined by parameter $flag$ in the algorithm description. If it is equal to 1, the algorithm is robust to one third errors in share distribution, otherwise only passive attack is considered. Concerning data representation, Shamir's secret sharing schemes works with integer numbers modulo a prime. Thus, a sufficiently large finite field F is selected to represent all the values in the modular domain $[0, p-1]$. Floating point numbers can be encoded as integers by simply multiplying them with same scale factor and the negative numbers can be represented with modular additive inverse. A rounding operation is needed in step 4 of Algorithm 1 to make sure all the input values in Shamir's secret sharing are integers.

V. Analysis

A comprehensive comparison of the proposed approach with existing approaches is shown in Table I, where β denotes the

Algorithm 2 Secure addition using Shamir's secret sharing

Polynomial construction:

All nodes $i \in C_i$ agree a polynomial degree t based on adversary model (active or passive).
Each node p_i randomly choose coefficients $c_i^1, c_i^2, ..., c_i^t$ on F, construct polynomial $f_i(x) = a_i + c_i^1 x + c_i^2 x^2 + ... + c_i^t x^t \mod p$.

Input sharing:

Each node p_i computes shares $f_i(j)$ and distributes shares $f_i(j)$ to all other nodes $\{j \mid j \in C_i, j \neq i\}$, respectively.
Each node p_i receives shares $f_j(i)$ from all other nodes $j \in C_i, j \neq i\}$, respectively.
Each node p_i computes sum l_i based on received shares $l_i = \sum_{j=1}^{n} f_j(i)$.
Each node p_i broadcasts l_i.

Output construction:

If $flag = 1$ (active attack model)
  Each node p_i defines $q(x)$ and $e(x)$ (see Section III).
  Each node p_i computes $q(x)$ and $e(x)$ based on share set $\{(i, l_i), i \in C_i\}$ with (4) and the desired polynomial $f(x)$ is determined by $q(x)/e(x)$.
  Each node p_i computes the result $y = f(0)$.
Else (passive attack model)
  Each node p_i computes r_i using (3).
  Each node p_i computes the result $y = \sum_{i \in C_i} r_i l_i$.
End

number of bits needed to represent encrypted cipher text [34]. We can see that the HE and GC approaches are computationally expensive and require high communication bandwidths, as the cipher texts after encryption usually require much longer bit length than plain texts. With the application of Shamir's secret sharing, the proposed algorithm outperforms differential privacy based approaches by having perfect security and identical accuracy with the non-privacy concerned algorithms [3]–[7]. Moreover, the involved functions are simpler than HE and GC based approaches and no trusted third party is required. The proposed approach also addresses a more challenging adversary model than the other approaches. For each iteration, where a node i is activated with n_i nodes in its clique, the proposed approach needs extra communication times compared to the other approaches because of the share distribution process.

A. Security analysis under passive and active attack

There is a difference between the privacy concern from a cryptographic point of view and a practical point of view. From the cryptographic point of view, the security definition imposes a very strong demand, namely, that a protocol is only secure if the adversary does not learn more information about the inputs of the honest nodes than what is implied by the output and the inputs of the corrupted nodes. From this point of view, the computation in each clique (see Algorithm 1), when considered in isolation, is information-theoretic (i.e., perfect) secure [29], but the full computation in Algorithm 1 would not be considered secure, since the adversary can learn the partial sums of the honest nodes' initial state values in some cliques, and this is not implied by the average result of the full network and the corrupted nodes' initial state values. However, from a practical point of view, as already stated in Section II-B, we are trying to protect the individual node's private information, and each individual node's initial state value is not revealed even if the sum of them are known.

For passive attacks, the privacy of the honest node will be protected as long as it has one honest neighbour in its clique. In a weaker model of active attacks, where the nodes act honestly when distributing the shares, but errors (either intentionally or unintentionally) can occur later on, the proposed algorithm can successfully reconstruct the correct result as long as at most one third of the shares are erroneous. This model captures cases such as unintentional errors produced when exchanging information. To the best of our knowledge, this is the first algorithm that obtains robustness against active attack in privacy-preserving distributed average consensus computation with both error detection and correction.

B. Security analysis under dynamic participation

One possible concern here is whether a clever combination of the information obtained in successive iterations can help to infer the privacy of the individual honest nodes, similarly to the privacy analysis in a dynamic setting [35] where nodes may come and leave between executions. This is, however, difficult to analyze. Note that since the inputs of the nodes involved in different iterations are dynamically updated, this is different from the case considered in [35] wherein inputs are static. We remark that it is difficult (without additional knowledge) for a passive adversary to ascertain whether any two iterations are successively related due to the random nature of the node activation and the clique selection in Algorithm 1. It is, however, possible that such situations can occur, and future research should investigate this further.

VI. Conclusions and Future Work

In this paper, we proposed a privacy-preserving distributed averaging consensus algorithm based on Shamir's secret sharing to compute the consensus in a distributed manner over a clique based network while protecting the individual privacy. The proposed algorithm is able to achieve both accurate consensus and perfect security at the same time, it does not depend on any trusted third party, and the computational complexity is lightweight. The adoption of Shamir's secret sharing allows to maintain the privacy of each individual node, i.e., as long as the clique selected in an iteration has at least 2 honest nodes. Moreover, robustness against up to one third errors is obtained under an active attack model. A drawback of the proposed approach is the higher communication times required by Shamir's secret sharing compared to, for example, differential privacy based methods. Future work will focus on how to reduce the overall communication times.

TABLE I
COMPARISONS WITH EXISTING APPROACHES

	Proposed	HE [22], [23]	GC [20]	Differential privacy [13]–[17]
Accuracy	Identical	Identical	Dependent on step size	Degraded with noise
Security	Perfect	Computational	Computational	Differential privacy
Attack model	Passive/Active	Passive	Passive	Passive
Involved function	Polynomial	Exponential	Exponential	Linear
Trusted Third Party	No	Yes	No	No
Communication bandwidth per time	$\mathcal{O}(1)$	$\mathcal{O}(\beta)$	$\mathcal{O}(\beta)$	$\mathcal{O}(1)$
Communication times per iteration	$\mathcal{O}(n_i^2)$	$\mathcal{O}(n_i)$	$\mathcal{O}(n_i)$	$\mathcal{O}(n_i)$

REFERENCES

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

[2] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *J. Parallel and Distributed Comput.*, vol. 7, no. 2, pp. 279–301, 1989.

[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] S. Aliaksei and K. Soummya and M. José MF, "Finite-time distributed consensus through graph filters," *ICASSP*, pp. 1080–1084, 2014.

[9] S. Santiago and M. Antonio G and R. Alejandro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, 2017.

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

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

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

[13] 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.

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

[15] 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.

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

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

[18] 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.

[19] 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.

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

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

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

[23] 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.

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

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

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

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

[28] C. Liu, "Introduction to combinatorial mathematics," 1968.

[29] A. Shamir, "How to share a secret," *Comm. Assoc. Comput. Mach.*, vol. 22, no. 11, pp. 612–613, 1979.

[30] L. R. Welch and E. R. Berlekamp, "Error correction for algebraic block codes," Dec. 1986.

[31] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," *FOCS*, pp. 383–395, 1985.

[32] S. Goryczka and L. Xiong, "A comprehensive comparison of multiparty secure additions with differential privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 5, pp. 463–477, 2017.

[33] S. C. S. Cheung and T. Nguyen, "Secure multiparty computation between distrusted networks terminals," *EURASIP J. Inf. Security*, vol. 2007, no. 1, pp. 051368, 2007.

[34] 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.

[35] D. Kononchuk, Z. Erkin, J. C. van der Lubbe, and R. L. Lagendijk, "Privacy-preserving user data oriented services for groups with dynamic participation," in *ESORICS*. pp. 418–442, 2013.