Privacy-preserving Average Consensus Algorithm with Beaver Triple

Peng Wang[1], Yang Lu[2], Jianming Lian[3], Lulu Pan[1], Haibin Shao[1], and Ning Li[1]

***Abstract*— A privacy-preserving average consensus algorithm is designed based on the Beaver triple technique against passive adversaries. The Beaver triple technique is integrated into a restructure of the discrete-time average consensus algorithm to preserve the privacy of initial values of agents in a multi-agent system. The performance of the algorithm is theoretically analyzed and simulation results with a power system application show the efficacy of the proposed algorithm.**

I. Introduction

Average consensus plays a pivotal role in various distributed applications, such as distributed control, estimation, and optimization. In these applications, a group of agents engages in iterative interactions with their neighbors within the communication graph. Through these interactions, agents update their states by assimilating information from neighboring agents, ultimately converging asymptotically to the average of all agents' initial states.

In traditional average consensus schemes, agents directly exchange their states with neighbors. However, this approach raises privacy concerns since sensitive information of legitimate agents may be exposed to adversarial agents or external eavesdroppers. Therefore, it becomes imperative to develop privacy-preserving algorithms capable of achieving average consensus while safeguarding the privacy of legitimate agents.

A. Literature Review

Privacy-preserving average consensus has garnered significant attention, with existing works falling into three main categories.

The first group utilizes differential privacy [1]–[5]. Here, agents introduce carefully crafted noises (often drawn from specific probability distributions, e.g., Gaussian and Laplace) into their states. This ensures that sensitive information, such as individual agents' initial states, cannot be deduced from the perturbed data, even with access to arbitrary auxiliary information. However, there exists a fundamental trade-off between privacy and accuracy in differentially private schemes due to the incorporation of random noises.

The second group employs encryption techniques to enable algebraic operations on ciphertexts while preserving plaintext privacy. Examples include homomorphic encryption, garbled circuit, and proxy re-encryption [6]–[12]. However, encryption schemes often involve modular exponentiation operations over large integers, leading to significant online computational overheads.

The third group focuses on obfuscating exchanged states by adding decaying or correlated noise and perturbation [13]–[20]. This approach ensures consensus accuracy, with works like [13]–[15] obscuring exchanged states to the extent that individual agents' initial states cannot be uniquely inferred. However, there remains a possibility of determining an interval of an agent's initial state, leading to potential privacy leakage. On the other hand, works such as [16], [17] propose obfuscation mechanisms that guarantee statistical privacy, mitigating the issue of interval leakage. Nevertheless, these approaches require that colluding adversarial agents do not form a vertex cut, which may pose constraints in certain real-world applications.

B. Contribution Statement

In this paper, we propose a novel privacy-preserving average consensus algorithm, called the Beaver-tripled-based average consensus (BAC) algorithm, that can simultaneously address the issues mentioned above.

Specifically, the Beaver triple provides a framework that splits the agents' private inputs and the edge weights into shares and restructures the computation in average consensus as a joint computation of the associated shares. This restructured joint computation is equivalent to the original computation, ensuring perfect computational correctness. Additionally, it prevents the agents' private inputs from being inferred during the joint computation.

C. Organization

The remaining sections of this paper are organized as follows. Section II provides preliminary knowledge. The privacy-preserving BAC algorithm is then designed in Section III, followed by an analysis of its privacy performance in Section IV. In Section V, a simulation is performed for a transactive energy system problem to demonstrate effectiveness of the proposed algorithm. Finally, the paper concludes in Section VI.

II. Preliminary

In this section, we introduce some preliminary knowledge of graph theory, distributed average consensus, and the Beaver triple technique.

[1]Peng Wang, Lulu Pan, Haibin Shao, and Ning Li are with the Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China. Emails: {wangpeng605, llpan, shore, ning_li}@sjtu.edu.cn.

[2]Yang Lu is with the School of Computing and Communications at the Lancaster University. Email: y.lu44@lancaster.ac.uk

[3]Jianming Lian is with Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA. Email: lianj@ornl.gov.

A. Graph Theory

An n-th order bidirectional graph, denoted as $\mathcal{G}(V, E, W)$, is defined by a vertex set $V = \{1, \cdots, n\}$, an edge set $E \subseteq V \times V$, and a weight matrix W. Each edge (i, j) represents a connection from vertex j to vertex i. It is assumed that $(i, i) \notin E$ for all $i \in V$. The weight matrix $W = (w_{ij})_{n\times n} \in \mathbb{R}^{n\times n}$ associated with the graph $\mathcal{G}$ is defined such that w_{ij} is positive if $(i, j) \in E$ or if $i = j$, and $w_{ij} = 0$ otherwise. In a bidirectional graph, the presence of $(i, j) \in E$ implies $(j, i) \in E$ as well. Additionally, W is row stochastic, ensuring $\sum_{j=1}^{n} w_{ij} = 1$ and $w_{ij} \geq 0$ for all $i \in V$, and symmetric, i.e., $w_{ij} = w_{ji}$ for all $i, j \in V$. The neighbor set of vertex i, denoted as N_i, consists of vertices j for which $(i, j) \in E$. A path between vertices i and j comprises a sequence of edges $(i, i_1), (i_1, i_2), \cdots, (i_p, j)$ connecting i to j. The bidirectional graph $\mathcal{G}$ is connected if, for any pair of vertices i and j, there exists a path connecting them.

B. Average Consensus Algorithm

A standard discrete-time consensus algorithm is represented by the following equation:

$$x_i(k+1) = \sum_{j=1}^{n} w_{ij}(k)x_j(k), \tag{1}$$

where $x_i(k)$ denotes the state of agent i at time step k, and $w_{ij}(k)$ represents the edge weight assigned by agent i to the edge (i, j). For simplicity, we assume scalar values for the agents' states $x_i(k)$ for all $i \in V$ and $k \in \mathbb{N}$. Denote $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$. The consensus algorithm in (1) can be equivalently rewritten as:

$$\boldsymbol{x}(k+1) = W(k)\boldsymbol{x}(k),$$

where $W(k) = (w_{ij}(k))_{n\times n} \in \mathbb{R}^{n\times n}$ represents the weight matrix at time step k.

The following assumptions are commonly made in the literature on multi-agent systems:

Assumption 1. *The communication graph $\mathcal{G}$ of the multi-agent system is connected.*

Assumption 2. *The weight matrix $W(k)$, for all $k \in \mathbb{N}$, is symmetric and row stochastic.*

Assumption 3. *There exist positive $\overline{w}_{ij}$ and $\underline{w}_{ij}$, for all $i, j \in V$, such that $w_{ij}(k) \in [\underline{w}_{ij}\ \overline{w}_{ij}]$ for all $k \in \mathbb{N}$ if $w_{ij}(k) \neq 0$.*

With these assumptions, we have the following lemma:

Lemma 1. *[21] Under Assumptions 1, 2, and 3, all agents in a multi-agent system asymptotically achieve an average consensus according to equation* (1)*, i.e., $\lim_{k\to\infty} x_i(k) = \frac{\sum_{j=1}^{n} x_j(0)}{n}$ for all $i \in V$.*

Remark 1. *From Assumptions 2 and 3, along with the condition that $(i, i) \notin E$, it follows that*

$$\sum_{\substack{j=1\\ j\neq i}}^{n} w_{ij}(k) = 1 - w_{ii}(k) < 1, \quad \forall k \in \mathbb{N}. \tag{2}$$

C. Beaver Triple

The Beaver triple, as introduced in [22], facilitates the computation of the product of shares held by two parties while maintaining the secrecy of those shares. Consider two agents, denoted as P_1 and P_2, each holding private values x_1, y_1 and x_2, y_2, respectively. To compute the product of $(x_1+x_2)(y_1+y_2)$ in a secret way, Beaver triples $a_i, b_i, c_i, i = 1, 2$ are first generated satisfying the condition $c_1 + c_2 = (a_1 + a_2)(b_1 + b_2)$. Define $a = a_1 + a_2$, $b = b_1 + b_2$, $c = c_1 + c_2$, $x = x_1 + x_2$, and $y = y_1 + y_2$. The computation unfolds in the following steps [22], [23]:

1) Each party $P_i, i \in \{1, 2\}$, computes the pair $(x_i - a_i, y_i - b_i)$ and sends the pair to the other party.

2) Each party $P_i, i \in \{1, 2\}$, computes $x - a = \sum_{j=1}^{2}(x_j - a_j)$, $y - b = \sum_{j=1}^{2}(y_j - b_j)$ using the received pair.

3) Each party $P_i, i \in \{1, 2\}$, computes $z_i = (x - a)b_i + (y - b)a_i + c_i$ and sends it to the other party.

4) Each party $P_i, i \in \{1, 2\}$, computes $xy = z_1 + z_2 + (x - a)(y - b)$.

Remark 2 (Motivating Example on Safeguarding Initial Values of Agents)**.** *In a transactive energy system (TES) [24], [25], average consensus algorithms can be used to achieve market clearing, in which the power[1] of all agents sums up to zero, or equivalently, averages at zero. The initial values of agents are their demand or supply curves representing their personal preferences for power at various prices. The agents will be at a disadvantage and may suffer losses if the true initial values are disclosed to others. Therefore, the privacy of the initial values should be preserved to encourage agents to actively participate in the TES.*

III. Beaver-triple-based Average Consensus Algorithm

A privacy-preserving distributed algorithm for average consensus, named the Beaver-triple-based average consensus (BAC) algorithm is designed in this section. The BAC algorithm integrates the Beaver triple technique into a reformulation of the average consensus algorithm in (1) to safegurad the initial values of agents in a multi-agent network.

First, the average consensus algorithm in (1) can be reformulated as

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{n} w_{ij}(k)x_j(k) \\ &= x_i(k) + \sum_{j\in N_i} w_{ij}(k)\left(x_j(k) - x_i(k)\right). \end{aligned} \tag{3}$$

[1]The convention is taken that the power supply is positive and the power demand is negative.

In (3), neighboring agents i and j need to collaboratively compute the message

$$m_{ij}(k) = w_{ij}(k)\left(x_j(k) - x_i(k)\right) \tag{4}$$

to update their states. Denote

$$\Delta x_{ij}(k) = x_j(k) - x_i(k). \tag{5}$$

From the perspective of secret sharing, $\Delta x_{ij}(k)$ can be regarded as a secret that is split into two shares $x_j(k)$ and $-x_i(k)$ owned by agent j and agent i, respectively. However, the privacy of $x_j(k)$ cannot be preserved against agent i because agent i can easily infer $x_j(k)$ though

$$m_{ij}(k)/w_{ij}(k) + x_i(k) \tag{6}$$

and vice versa agent j can infer the value of $x_i(k)$.

One way to preserve the privacy of agents' states is to make the edge weight $w_{ij}(k)$ also a secret that is split into two shares $w_{ij,i}(k)$ and $w_{ij,j}(k)$ owned by agents i and j, respectively, with

$$w_{ij}(k) = w_{ij,i}(k) + w_{ij,j}(k). \tag{7}$$

Intuitively, agent i cannot infer the value of $x_j(k)$ through (6) if it does not know $w_{ij,j}(k)$.

With (5) and (7), the message $m_{ij}(k)$ in (4) becomes the product of two secrets, i.e., $\Delta x_{ij}(k)$ in (5) and $w_{ij}(k)$ in (7), each of which is split into two shares owned by agents i and j, respectively, shown as follows

$$m_{ij}(k) = \left(w_{ij,i}(k) + w_{ij,j}(k)\right)\left(-x_i(k) + x_j(k)\right), \tag{8}$$

where $w_{ij,i}(k)$ and $-x_i(k)$ are private shares of agent i, and $w_{ij,j}(k)$ and $x_j(k)$ are private shares of agent j. To preserve the privacy of $x_i(k)$ (or $x_j(k)$) against agent j (or agent i), the Beaver-triple-based multiplicative secret sharing technique is applied such that agents i and j cooperatively obtain m_{ij} without knowing the exact values of x_j and x_i, respectively.

The detailed steps to compute $m_{ij}(k)$ with the Beaver triple technique are summarized in Algorithm 1, in which the step index k is omitted and the superscript "+" in Steps 10 and 11 represents the next step.

Remark 3. *A key condition to the successful implementation of Algorithm 1 is to obtain the Beaver triples a_i, b_i, c_i and a_j, b_j, c_j by agents i and j without disclosing the triples to other parties. Methods in the literature, e.g., [26], can be used to generate multiple Beaver triples in such a way.*

Remark 4 (Range of Edge Weight)**.** *Bounds of weight shares $\underline{w}_{ij,i}$, $\overline{w}_{ij,i}$, $\underline{w}_{ij,j}$, $\overline{w}_{ij,j}$ are needed prior to running the algorithm to ensure that the selection of $w_{ij,i}$ and $w_{ij,j}$ satisfies* (2)*. A straightforward way to obtain such bounds is to make*

$$\underline{w}_{ij,i} = \underline{w}_{ij,j} = \frac{\underline{w}_{ij}}{2},$$
$$\overline{w}_{ij,i} = \overline{w}_{ij,j} = \frac{\overline{w}_{ij}}{2},$$

where $\underline{w}_{ij}$ and $\overline{w}_{ij}$ are specified in Assumption 3.

Algorithm 1 Beaver-triple-based Average Consensus (BAC) Algorithm

Input: Beaver triples a_i, b_i, c_i and a_j, b_j, c_j pre-computed in the offline process, the bounds of weight shares $\underline{w}_{ij,i}$, $\overline{w}_{ij,i}$, $\underline{w}_{ij,j}$, $\overline{w}_{ij,j}$ negotiated offline

Output: the message m_{ij}

1: agent i generates $w_{ij,i}$ in the interval $[\underline{w}_{ij,i}\ \overline{w}_{ij,i}]$
2: agent j generates $w_{ij,j}$ in the interval $[\underline{w}_{ij,j}\ \overline{w}_{ij,j}]$
3: agent i computes $x_i^{(a)} = -x_i - a_i$ and $w_i^{(b)} = w_{ij,i} - b_i$ and sends $x_i^{(a)}$ and $w_i^{(b)}$ to agent j
4: agent j computes $x_j^{(a)} = x_j - a_j$ and $w_j^{(b)} = w_{ij,j} - b_j$ and sends $x_j^{(a)}$ and $w_j^{(b)}$ to agent i
5: both agent i and agent j calculate $\Delta x_{ij}^{(a)} = x_j^{(a)} + x_i^{(a)}$ and $\Delta w_{ij}^{(b)} = w_j^{(b)} + w_i^{(b)}$, where $\Delta x_{ij}^{(a)}$ is $x_j - x_i - a$ and $\Delta w_{ij}^{(b)}$ is $w_{ij} - b$
6: agent i computes $z_i = \Delta x_{ij}^{(a)} b_i + \Delta w_{ij}^{(b)} a_i + c_i$ and sends it to agent j
7: agent j computes $z_j = \Delta x_{ij}^{(a)} b_j + \Delta w_{ij}^{(b)} a_j + c_j$ and sends it to agent i
8: agent i computes $m_{ij} = z_i + z_j + \Delta x_{ij}^{(a)} \Delta w_{ij}^{(b)}$
9: agent j computes $m_{ji} = -(z_i + z_j + \Delta x_{ij}^{(a)} \Delta w_{ij}^{(b)})$
10: agent i updates its state $x_i^+ = x_i + \sum_{j \in N_i} m_{ij}$
11: agent j updates its state $x_j^+ = x_j + \sum_{j \in N_i} m_{ji}$

Remark 5. *As the time-consuming and computation-extensive steps to generate the Beaver triples can be done offline prior to the iterative consensus process in Algorithm 1, the agent i only needs to compute $x_i^{(a)}$, $w_i^{(b)}$, $\Delta x_{ij}^{(a)}$, $\Delta w_{ij}^{(b)}$, z_i, and m_{ij} though summation and multiplication of real numbers in plaintexts and send $x_i^{(a)}$, $w_i^{(b)}$, and z_i that are all real numbers in plaintexts. Therefore, Algorithm 1 is more computationally and communicationally efficient for online implementation than encryption-based average consensus algorithms, e.g., [9].*

Remark 6 (Consensus of BAC Algorithm)**.** *One of the differences of Algorithm 1 from the typical average consensus algorithm in* (3) *is that the message m_{ij} is computed with the Beaver-triple-based multiplicative secret sharing technique. All agents' states $x_i(k), i \in V, k \in \mathbb{N}$, generated by Algorithm 1 and* (3) *are identical with identical initial conditions, identical communication topology, and identical edge weights. Therefore, all agents in Algorithm 1 reach an average consensus under Assumptions 1, 2, and 3 from Lemma 1.*

IV. Privacy Analysis

In this section, the privacy property of Algorithm 1 is analyzed.

For the adversary, it is assumed that

Assumption 4 (Passive Adversary)**.** *All adversarial agents are passive, or equivalently, honest but curious. They follow the rules of the designed algorithm but intend to infer the initial states of benign agents.*

The information that an adversary knows via Algorithm 1 is defined in the information set as follows.

Definition 1 (Information Set)**.** *For any initial conditions $C = \{x_E(0), x_j(0), j \in V/\{E\}\}$, the information set of agent E is*

$$I_E^{(C)} = \{x_E(k), m_{Ej}(k), j \in N_E, k \in \mathbb{N}\}. \tag{9}$$

for Algorithm 1, where N_E is the set of neighbors of the agent E.

Before analyzing the privacy property, we give definitions of bounded privacy, which represents different levels of privacy preservation degrees.

Definition 2 (Definition of Bounded Privacy)**.** *The bounded privacy of the initial value of an agent A against a passively adversarial agent E is preserved if, for any set of initial conditions $C = \{x_E(0), x_A(0), x_j(0), j \in V/\{A, E\}\}$, there exists $x'_A(0)$ in a neighborhood of $x_A(0)$ such that $I_E^{(C)} = I_E^{(C')}$, where $C' = \{x_E(0), x'_A(0), x'_k(0), k \in V/\{A, E\}\}$.*

Given an initial condition C, the bounded privacy in Definition 2 requires only the existence of one x'_A that generates the same information set.

The following assumption on communication topology regarding the neighborhood of benign agents is also made.

Assumption 5. *None of the adversaries is the unique neighbor of a benign agent.*

In general, a vicinity of the true state of an agent can be inferred by adversaries with only the protection of the Beaver triple, as stated in the following lemma.

Lemma 2. *An adversarial agent E can infer a range of the initial state of any of its neighboring agent A with the information of $m_{AE}(0)$.*

Proof: Denote

$$r_{AE}^{(l)}(0) = \min\{\frac{m_{AE}(0)}{w_{AE,E}(0) + \underline{w}_{AE,A}}, \frac{m_{AE}(0)}{w_{AE,E}(0) + \overline{w}_{AE,A}}\},$$
$$r_{AE}^{(u)}(0) = \max\{\frac{m_{AE}(0)}{w_{AE,E}(0) + \underline{w}_{AE,A}}, \frac{m_{AE}(0)}{w_{AE,E}(0) + \overline{w}_{AE,A}}\}.$$

From (8), we can then obtain that

$$\begin{aligned} x_A(0) =& x_E(0) - \frac{m_{AE}(0)}{w_{AE,A}(0) + w_{AE,E}(0)} \\ \in& [x_E(0) - r_{AE}^{(u)}(0),\ x_E(0) - r_{AE}^{(l)}(0)], \end{aligned}$$

where $\underline{w}_{AE,A}$ and $\overline{w}_{AE,A}$ are the minimum and maximum of $w_{AE,A}(0)$, respectively.

The interval specified in Lemma 2 is formally defined as follows:

Definition 3 (Inferred Interval)**.** *The inferred interval from m_{ij} by agent j is*

$$\left[x_j - r_{ij}^{(u)},\ x_j - r_{ij}^{(l)}\right], \tag{10}$$

where

$$r_{ij}^{(l)} = \min\{\frac{m_{ij}}{w_{ij,j} + \underline{w}_{ij,i}}, \frac{m_{ij}}{w_{ij,j} + \overline{w}_{ij,i}}\},$$
$$r_{ij}^{(u)} = \max\{\frac{m_{ij}}{w_{ij,j} + \underline{w}_{ij,i}}, \frac{m_{ij}}{w_{ij,j} + \overline{w}_{ij,i}}\}.$$

Remark 7. *If x'_i is in the inferred interval from m_{ij} by agent j, then there always exists $w'_{ij,i}$ such that $m_{ij} = (w'_{ij,i} + w_{ij,j})(x_j - x'_i)$.*

From Lemma 2, the privacy of the initial values of agents cannot be preserved by Algorithm 1 because the adversary can immediately learn a bounded interval for states of its neighboring agents. Next, we investigate whether bounded privacy can be preserved with the BAC algorithm.

Lemma 3. *Suppose that agent B is a neighbor of agent A. Under the conditions that*

1) there exists a number $\varepsilon \neq 0$ such that $x'_A(0) = x_A(0) + \varepsilon$ is in the inferred interval of $m_{AE}(0)$ by agent E;

2) if $B \in N_E$, $x'_B(0) = x_B(0) - \varepsilon$ is also in the inferred interval of $m_{BE}(0)$ by agent E;

3) $\frac{(w_{BA,A}(0) + w_{BA,B}(0))(x_B(0) - x_A(0)) - \varepsilon}{x_B(0) - x_A(0) - 2\varepsilon} \in [\underline{w}_{ij}\ \ \overline{w}_{ij}]$, where $\underline{w}_{ij}$ and $\overline{w}_{ij}$ are specified in Assumption 3;

there exists $w'_{AB,A}(0)$, $w'_{AB,B}(0)$, $w'_{EA,A}(0)$, and $w'_{EB,B}(0)$ such that $I_E^{(C)} = I_E^{(C')}$, where $C = \{x_E(0), x_A(0), x_j(0), j \in V/\{A, E\}\}$, $C' = \{x_E(0), x'_A(0) = x_A(0) + \varepsilon, x'_B(0) = x_B(0) - \varepsilon, x_j(0), j \in V/\{A, B, E\}\}$.

Proof: We consider two cases:

Case 1: agent B is a neighbor of agent E. Then, from (9), $I_E^{(C)} = \{x_E(k), m_{EA}(k), m_{EB}(k), m_{Ej}(k), j \in N_E/\{A, B\}, k \in \mathbb{N}\}$ and $I_E^{(C')} = \{x_E(k), m'_{EA}(k), m'_{EB}(k), m'_{Ej}(k), j \in N_E/\{A, B\}, k \in \mathbb{N}\}$, where $m_{EA}(k)$ and $m_{EB}(k)$ are defined as in (8), $x_E(k)$ the state of agent E, $m'_{EA}(k)$ and $m'_{EB}(k)$ are the counterpart of $m_{EA}(k)$ and $m_{EB}(k)$, respectively. Then, the following three conditions are sufficient to guarantee $I_E^{(C)} = I_E^{(C')}$:

a) $m_{EA}(0) = m'_{EA}(0)$,

b) $m_{EB}(0) = m'_{EB}(0)$,

c) $x_A(1) = x'_A(1)$ and $x_B(1) = x'_B(1)$

For the first condition, as $x'_A(0)$ is in the inferred interval of $m_{AE}(0)$ by agent E, from Remark 7, there exists a $w'_{EA,A}(0) \in (\underline{w}_{EA,A}, \overline{w}_{EA,A})$ such that

$$\begin{aligned} m_{EA}(0) &= (w_{EA,A}(0) + w_{EA,E}(0))(x_A(0) - x_E(0)) \\ &= (w'_{EA,A}(0) + w_{EA,E}(0))(x'_A(0) - x_E(0)) \\ &= m'_{EA}(0). \end{aligned} \tag{11}$$

By solving (11), it can be obtained that

$$\begin{aligned}&w'_{EA,A}(0)\\=&\frac{(w_{EA,A}(0)+w_{EA,E}(0))(x_A(0)-x_E(0))}{x'_A(0)-x_E(0)}-w_{EA,E}(0)\\=&\frac{m_{EA}(0)}{x'_A(0)-x_E(0)}-w_{EA,E}(0).\end{aligned}\tag{12}$$

For the second condition, it can similarly be obtained that

$$\begin{aligned}m_{EB}(0)&=(w_{EB,B}(0)+w_{EB,E}(0))(x_B(0)-x_E(0))\\&=(w'_{EB,B}(0)+w_{EB,E}(0))(x'_B(0)-x_E(0))\\&=m'_{EB}(0).\end{aligned}\tag{13}$$

and

$$\begin{aligned}&w'_{EB,B}(0)\\=&\frac{(w_{EB,B}(0)+w_{EB,E}(0))(x_B(0)-x_E(0))}{x'_B(0)-x_E(0)}-w_{EB,E}(0)\\=&\frac{m_{EB}(0)}{x'_B(0)-x_E(0)}-w_{EB,E}(0).\end{aligned}\tag{14}$$

For the third condition, first $x_A(1)=x'_A(1)$ and $x_B(1)=x'_B(1)$ are both equivalent to

$$m_{BA}(0)-m'_{BA}(0)=x'_A(0)-x_A(0)=\varepsilon.\tag{15}$$

For agent A, that is because (15) is equivalent to

$$x_A(0)-x'_A(0)+(m_{BA}(0)-m'_{BA}(0))=0,$$

which is equivalent to $x_A(1)=x'_A(1)$ with the fact that

$$\begin{aligned}x_A(1)&=x_A(0)+m_{BA}(0)+m_{EA}(0),\\x'_A(1)&=x'_A(0)+m'_{BA}(0)+m_{EA}(0).\end{aligned}$$

Similar analysis can be done for the condition $x_B(1)=x'_B(1)$ to obtain the following equivalent condition

$$m_{AB}(0)-m'_{AB}(0)=x'_B(0)-x_B(0)=-\varepsilon.$$

Under the condition that $m_{AB}(0)=-m_{BA}(0)$, $m'_{AB}(0)=-m'_{BA}(0)$, and $x'_B(0)-x_B(0)=-\varepsilon=-(x'_A(0)-x_A(0))$, it can be obtained that for agent B, $x_B(1)=x'_B(1)$ is also equivalent to (15).
From (15),

$$\begin{aligned}&(w_{BA,A}(0)+w_{BA,B}(0))(x_B(0)-x_A(0))\\&-(w'_{BA,A}(0)+w'_{BA,B}(0))(x'_B(0)-x'_A(0))\\=&x'_A(0)-x_A(0).\end{aligned}\tag{16}$$

Thus, if $x'_B(0)-x'_A(0)\neq 0$,

$$\begin{aligned}&w'_{BA,A}(0)+w'_{BA,B}(0)\\=&\frac{(w_{BA,A}(0)+w_{BA,B}(0))(x_B(0)-x_A(0))}{x'_B(0)-x'_A(0)}\\&-\frac{(x'_A(0)-x_A(0))}{x'_B(0)-x'_A(0)}\\=&\frac{m_{BA}-(x'_A(0)-x_A(0))}{x'_B(0)-x'_A(0)};\end{aligned}\tag{17}$$

if $x'_B(0)-x'_A(0)=0$, then (16) becomes $m_{AB}(0)=\varepsilon$ and $m'_{AB}(0)=0$, which is valid for any selection of $w'_{BA,A}(0)+w'_{BA,B}(0)$ from (15).

Therefore, if agent B is a neighbor of agent E, $I_E^{(C)}=I_E^{(C')}$.

Case 2: agent B is not a neighbor of agent E. Then, $I_E^{(C)}=\{x_E(k),m_{EA}(k),m_{Ej}(k),j\in N_E/\{A\},k\in\mathbb{N}\}$ and $I_E^{(C')}=\{x_E(k),m'_{EA}(k),m'_{Ej}(k),j\in N_E/\{A\},k\in\mathbb{N}\}$. The following conditions are sufficient to guarantee $I_E^{(C)}=I_E^{(C')}$:

a) $m_{EA}(0)=m'_{EA}(0)$,

b) $x_A(1)=x'_A(1)$ and $x_B(1)=x'_B(1)$.

Following a similar process in the analysis of (12) and (17), we can find $w'_{AB,A}(0)$, $w'_{AB,B}(0)$, and $w'_{EA,A}(0)$ making $I_E^{(C)}=I_E^{(C')}$.

In summary, there exist $w'_{AB,A}(0)$, $w'_{AB,B}(0)$, $w'_{EA,A}(0)$, and $w'_{EB,B}(0)$ such that $I_E^{(C)}=I_E^{(C')}$ whether agent B is a neighbor of the agent E or not under the conditions in this lemma.

In Lemma 3, the bounded privacy of Algorithm 1 is analyzed assuming the existence of ε, which is not yet proven. Next, we analyze the conditions on the existence of ε when the agent B is neighboring to the agent E in the following lemma.

Lemma 4 (existence of ε)**.** *Suppose that agent B is a neighbor of agent E. If $x_A(0)-x_E(0)\neq 0$, $x_B(0)-x_E(0)\neq 0$, $x_B(0)-x_A(0)\neq 0$, $w_{EA,A}(0)\in(\underline{w}_{EA,A},\ \overline{w}_{EA,A})$, $w_{EB,B}(0)\in(\underline{w}_{EB,B},\ \overline{w}_{EB,B})$, and $w_{BA}(0)\in(\underline{w}_{BA},\ \overline{w}_{BA})$, i.e., none of $w_{EA,A}(0),w_{EB,B}(0),w_{BA}(0)$ reaches its bounds, then the conditions 1), 2), and 3) in Lemma 3 hold.*

Proof: Define

$$\begin{aligned}f_1(\varepsilon,w'_{EA,A})&=(w'_{EA,A}+w_{EA,E})(x'_A-x_E)-m_{EA}\\&=(w'_{EA,A}+w_{EA,E})(x_A-x_E+\varepsilon)-m_{EA},\\f_2(\varepsilon,w'_{EB,B})&=(w'_{EB,B}+w_{EB,E})(x'_B-x_E)-m_{EB}\\&=(w'_{EB,B}+w_{EB,E})(x_B-x_E-\varepsilon)-m_{EB},\\f_3(\varepsilon,w'_{BA})&=(w'_{BA,A}+w'_{BA,B})(x'_B-x'_A)-m_{BA}+\varepsilon\\&=w'_{BA}(x_B-x_A-2\varepsilon)-m_{BA}+\varepsilon,\end{aligned}$$

and

$$F(\varepsilon,w'_{EA,A},w'_{EB,B},w'_{BA})=(f_1,\ f_2,\ f_3)^T.$$

It can be obtained that

$$F(0,w_{EA,A},w_{EB,B},w_{BA})=0\tag{18}$$

from (11), (13), and (16).

For simplicity of notations, denote $y=(w'_{EA,A},w'_{EB,B},w'_{BA})^T$. Then, the Jacobian J_y of F with respect to y is a diagonal matrix with the diagonal

entries being

$$J_{y,11} = \frac{\partial f_1}{\partial w'_{EA,A}} = x_A - x_E + \varepsilon,$$
$$J_{y,22} = \frac{\partial f_2}{\partial w'_{EB,B}} = x_B - x_E - \varepsilon,$$
$$J_{y,33} = \frac{\partial f_3}{\partial w'_{BA}} = x_B - x_A - 2\varepsilon.$$

Under the condition that $x_A - x_E \neq 0$, $x_B - x_E \neq 0$, and $x_B - x_A \neq 0$,

$$\det J_y|_{\varepsilon=0} \neq 0, \tag{19}$$

where $\det$ represents the determinant of a matrix.

Combining (18) and (19) with the facts that F is well-defined and continuous in a neighborhood of $(0, w_{EA,A}, w_{EB,B}, w_{BA})$ and that its partial derivatives with respect y exist and are continuous, it can be obtained from the implicit function theorem [27] that in a neighborhood of $(0, w_{EA,A}, w_{EB,B}, w_{BA})$, y, or equivalently, $w'_{EA,A}, w'_{EB,B}, w'_{BA}$, can be expressed as continuous functions of ε with continuous derivatives. Therefore, if $w_{EA,A}, w_{EB,B}, w_{BA}$ are not at their lower or upper bounds, there exists $\varepsilon \neq 0$ such that the conditions (1), (2), and (3) in Lemma 3 hold.

When agent B is not a neighbor of agent E, following a similar process to the proof of Lemma 4, we can obtain the following lemma on the existence of ε:

Lemma 5. *Suppose that agent B is not a neighbor of agent E. If $x_A(0) - x_E(0) \neq 0$ and $x_B(0) - x_A(0) \neq 0$, $w_{EA,A}(0) \in (\underline{w}_{EA,A}, \overline{w}_{EA,A})$, $w_{EB,B}(0) \in (\underline{w}_{EB,B}, \overline{w}_{EB,B})$, and $w_{BA}(0) \in (\underline{w}_{BA}, \overline{w}_{BA})$, i.e., none of $w_{EA,A}(0), w_{EB,B}(0), w_{BA}(0)$ reaches its bounds, then the conditions 1) and 3) in Lemma 3 hold.*

Combining Lemmas 3, 4, and 5, we can obtain that the BAC algorithm in Algorithm 1 can preserve bounded privacy of the initial values of agents against passive adversaries. It results from the bounds of the edge weights. To improve the privacy preservation performance of the BAC algorithm, agent i can generate $w_{ij,i}$ from $(-\infty, \infty)$ at the initial step and then from the range specified in Algorithm 1 at the rest steps. With such a selection, the BAC algorithm is computationally equivalent to

$$x_i(1) = \sum_{j=1}^{n} w_{ij}(0)x_j(0),\ w_{ij}(k) \in \mathbb{R}$$

at the initial step. From the symmetry of the weight matrix $W(0)$, it can be obtained that

$$\sum_{j=1}^{n} x_i(1) = \sum_{i=1}^{n} x_i(0)$$

and thus the average consensus can be achieved. In this way, we can concurrently preserve the unbounded privacy of the initial values of agents and reach an average consensus.

Remark 8. *The analysis in this subsection focuses on the interaction among the agents A, B, and E. When there is another agent C connected to the agent A, a similar analysis to that in Lemmas 3 and 4 can also be performed with $x'_C(0) = x_C(0)$ and Condition 3) in Lemma 3 should also include the one obtained considering the agent C.*

V. Simulation

In this section, the proposed BAC algorithm is verified through simulation on a TES consisting of 1000 consumers represented by controllable air conditioners (ACs) and one supplier representing the utility. The market period is selected to be five minutes. The lowest market-clearing price $\lambda_{\min}$ is set to be \$0.00, the highest price $\lambda_{\max} = \$1.00$, and the price increment $\Delta\lambda = \$0.01$.

The supply curve of the supplier is set to be $p^s(\lambda) = P^s_{\max}\lambda^{\frac{1}{3}}$, where $P^s_{\max}$ is the supplier's capacity, and λ is the price between \$0.00 and \$1.00. $P^s_{\max}$ is set to be the sum of the rated power of all ACs, which makes the supplier able to provide enough energy to all ACs. The controllable load of consumers is residential ACs. The rated power of the ACs is uniformly distributed between 2.5 kW and 5.0 kW. The thermal parameters of buildings, which are related to floor area, ceiling height, glass type, glazing layers, material, area per floor, etc., are derived from GridLAB-D, of which the details can be found in [28]. The weather data and the Typical Meteorological Year (TMY) data are obtained for Columbus, OH from [29] and [30], respectively.

A. Demand Curve Generation

The demand curve for a residential AC is generated in this subsection. The occupants of a house directly control the temperature setpoint, instead of the power consumption, of the AC. Thus, the demand curve is generated in the following two steps: 1) a response curve is generated to characterize the relationship between the price and AC temperature set point; 2) the relationship between the AC temperature setpoint and the AC power consumption is established.

The response curve is generated as follows. The house occupants specify their comfort zones characterized by minimum temperature set point T_{min}, desired temperature set point T_{desired}, and maximum temperature set point T_{max}. ACs compute the average λ_{avg} and variance σ of the historical market-clearing price over a period of time in the past, e.g., 24 hours. The control response curve is then generated as shown in Fig. 1. The parameter k in Fig. 1, which is related to the occupant's tolerance to price change and indoor temperature change, can be obtained from their historical behavior. When the price is the average of the historical price, the temperature set point is the desired one. The temperature set point is responsive to the price when the price is between $\lambda_{\text{avg}} - k\sigma$ and $\lambda_{\text{avg}} + k\sigma$ and becomes irresponsive when the price is out of the range.

After the temperature setpoint T_{set} is selected, the power consumption of the AC is computed from the AC dynamics in (20) and the hysteresis temperature control. The dynamics

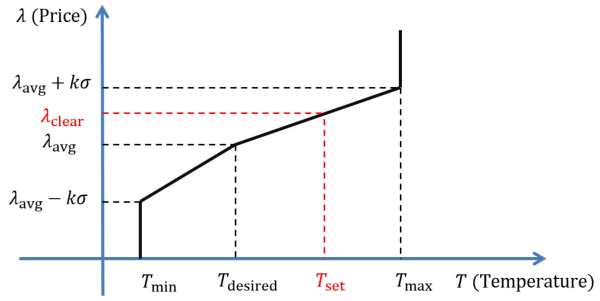

Fig. 1. Illustration of control response curve [31]

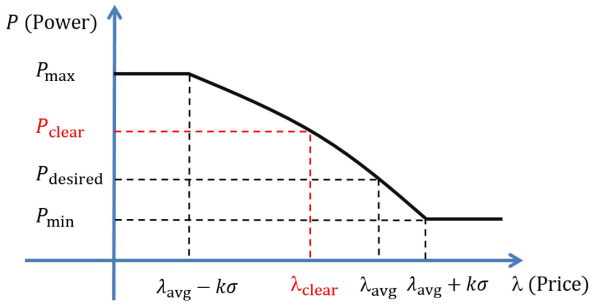

Fig. 2. Illustration of the AC demand curve [31].

of the AC are [31],

$$\begin{aligned}\dot{T}_a &= \frac{H_m}{C_a}(T_m - T_a) + \frac{U_a}{C_a}(T_o - T_a) + \frac{Q}{C_a},\\ \dot{T}_m &= \frac{H_m}{C_m}(T_a - T_m) + \frac{Q}{C_m},\end{aligned} \tag{20}$$

where T_a is the indoor air temperature, T_m is the inner mass temperature, e.g. that of the building materials and furnishings, U_a is the conductance of the building envelope, T_o is the outdoor air temperature, H_m is the conductance between the inner air and inner solid mass, C_a is the thermal mass of the air, C_m is the thermal mass of the building materials and furnishings, Q is the total heat flux consisting of the heat gain from the internal load Q_i , the solar heat gain Q_s and the heat gain from the cooling system Q_h, and

$$Q(t) = \begin{cases} Q_i + Q_s + Q_h, & \text{when AC is on,}\\ Q_i + Q_s, & \text{when AC is off.}\end{cases}$$

The hysteresis temperature control over a given temperature set point T_{set} and predefined temperature deadband δ is as follows: the AC is turned on if the indoor air temperature is higher than $T_{\text{set}} + \delta/2$, is turned off if the indoor air temperature is lower than $T_{\text{set}} - \delta/2$, and keeps the previous on/off state if the indoor air temperature is between $T_{\text{set}}+\delta/2$ and $T_{\text{set}} - \delta/2$.

The power demand corresponding to a temperature set point is then calculated as the product of the AC rated power and the period of AC being on, where the rated power is proportional to the heat gain Q_h from the cooling system. The demand curve can then be obtained from a combination of the response curve and the temperature setpoint-power relationship. The resulting demand curve is illustrated in Fig. 2. As λ_{avg}, σ, and the power demand, which depends on indoor air temperature, outdoor air temperature, solar heat gain, etc., changes with time, the demand curves of ACs are also time-varying.

B. BAC-based Market-clearing Results

A simulation for 24-hour market clearing with the BAC algorithm is performed. The market-clearing results with the centralized market-clearing method and the proposed BAC algorithm are presented in Fig. 3. With the centralized method, all suppliers and consumers submit their supply curves and demand curves, respectively, to a coordinator and

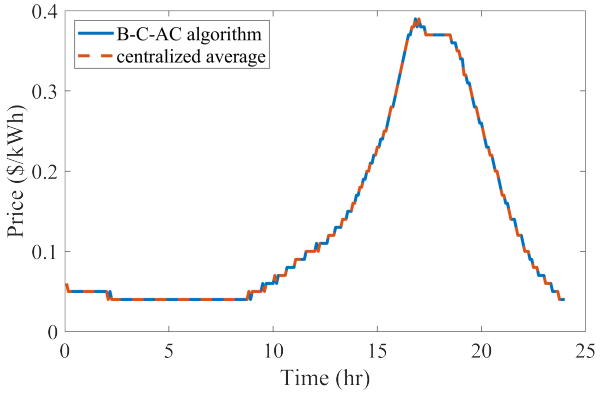

Fig. 3. Market-clearing prices with the centralized method and the BAC algorithm

the coordinator then clears the market at the price point at which the power supply equals the power demand. With the BAC algorithm, all suppliers and consumers collaborate to clear the market. From Fig. 3, it can be observed that the BAC algorithm can ensure the correct market-clearing results for the TES.

The consensus error of the power of each supplier and consumer at the market-clearing price at a randomly selected market period is presented in Fig. 4. The curve starting from a very large positive value represents the supplier while all other curves represent consumers. As the capacity of the supplier covers the power consumption of all consumers, the initial value, which is the power supply or demand preference at the market-clearing price, of the supplier is much larger than each consumer. Fig. 4 is zoomed in in Fig. 5 to show the power of consumers in the consensus process. From Figs. 4 and 5, it can be observed that the power of all suppliers and consumers converges to the average of their initial power.

VI. Conclusion

A Beaver-triple-based average consensus (BAC) algorithm was proposed to preserve the privacy of the initial values of agents against passive adversaries in a multi-agent system. The algorithm is proven to protect the privacy of a benign agent against passive adversaries if none of the adversaries is the unique neighbor of the agent in the multi-agent network. A simulation example of a transactive energy system is adopted to show the efficacy of the BAC algorithm. Future

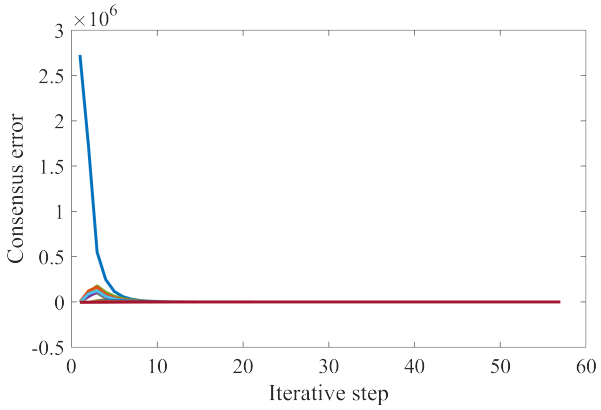

Fig. 4. Consensus error of the market-clearing power to the average of initial power at the market-clearing price

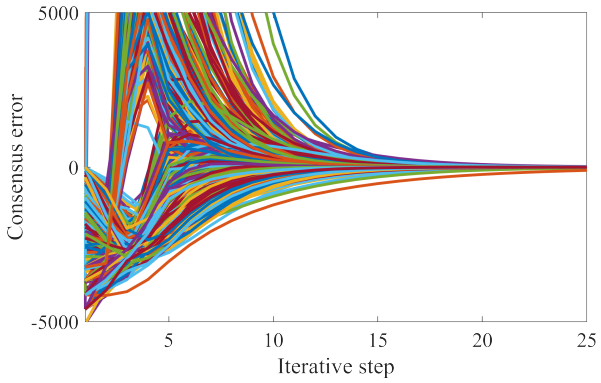

Fig. 5. Consensus error of the market-clearing power to the average of initial power at the market-clearing price (zoomed in)

work may be focused on extending the results to other problems, e.g., distributed optimization and multi-agent learning.

REFERENCES

[1] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, pp. 81–90, 2012.

[2] J. He, L. Cai, and X. Guan, "Differential Private Noise Adding Mechanism and Its Application on Consensus Algorithm," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4069–4082, 2020.

[3] 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.

[4] C. Zhao, J. Chen, J. He, and P. Cheng, "Privacy-Preserving Consensus-Based Energy Management in Smart Grids," *IEEE Transactions on Signal Processing*, vol. 66, no. 23, pp. 6162–6176, 2018.

[5] L. Gao, S. Deng, and W. Ren, "Differentially private consensus with an event-triggered mechanism," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 60–71, 2019.

[6] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *Proceedings of 2015 IEEE 54th Annual Conference on Decision and Control (CDC)*, pp. 6836–6843, December 2015.

[7] Y. Lu and M. Zhu, "Privacy preserving distributed optimization using homomorphic encryption," *Automatica*, vol. 96, pp. 314–325, October 2018.

[8] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *Proceedings of the 2016 IEEE 55th Conference on Decision and Control*, pp. 5053–5058, December 2016.

[9] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Trans. Automat. Contr.*, vol. 64, no. 10, pp. 4035–4049, 2019.

[10] W. Fang, M. Zamani, and Z. Chen, "Secure and privacy preserving consensus for second-order systems based on paillier encryption," *Syst. Control Lett.*, vol. 148, no. 104869, p. 104869, 2021.

[11] M. Ambrosin, P. Braca, M. Conti, and R. Lazzeretti, "Odin: O bfuscation-based privacy-preserving consensus algorithm for d ecentralized i nformation fusion in smart device n etworks," *ACM Transactions on Internet Technology (TOIT)*, vol. 18, no. 1, pp. 1–22, 2017.

[12] R. Lazzeretti, S. Horn, P. Braca, and P. Willett, "Secure multi-party consensus gossip algorithms," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7406–7410, 2014.

[13] Y. Mo and R. M. Murray, "Privacy Preserving Average Consensus," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 753–765, 2017.

[14] S. Gade and N. H. Vaidya, "Privacy-preserving distributed learning via obfuscated stochastic gradients," in *2018 IEEE Conference on Decision and Control*, pp. 184–191, 2018.

[15] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed Privacy-Preserving Data Aggregation Against Dishonest Nodes in Network Systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1462–1470, 2019.

[16] N. Gupta, J. Katz, and N. Chopra, "Privacy in distributed average consensus," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9515–9520, 2017.

[17] N. Gupta, J. Kat, and N. Chopra, "Statistical privacy in distributed average consensus on bounded real inputs," in *2019 American Control Conference (ACC)*, IEEE, 2019.

[18] N. E. Manitara and C. N. Hadjicostis, "Privacy-preserving asymptotic average consensus," in *2013 European Control Conference (ECC)*, pp. 760–765, 2013.

[19] Y. Xiong and Z. Li, "Privacy-preserved average consensus algorithms with edge-based additive perturbations," *Automatica*, vol. 140, p. 110223, 2022.

[20] C. Altafini, "A system-theoretic framework for privacy preservation in continuous-time multiagent dynamics," *Automatica*, vol. 122, p. 109253, 2020.

[21] W. Ren and R. Beard, *Distributed consensus in multi-vehicle cooperative control: Theory and applications*. London, England: Springer, 2010.

[22] D. Evans, V. Kolesnikov, and M. Rosulek, *A pragmatic introduction to secure multi-party computation*. Hanover, MD: now, 2018.

[23] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Advances in Cryptology — CRYPTO '91*, pp. 420–432, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[24] J. Lian, W. Zhang, Y. Sun, L. D. Marinovici, K. Kalsi, and S. E. Widergren, "Transactive system: Part I: Theoretical underpinnings of payoff functions, control decisions, information privacy, and solution concepts," Tech. Rep. PNNL-27235 Part I, Pacific Northwest National Laboratory, 2018.

[25] J. Lian, Y. Sun, D. Wu, H. Ren, K. Kalsi, and S. E. Widergren, "Transactive system: Part II: Analysis of two pilot transactive systems using foundational theory and metrics," Tech. Rep. PNNL-27235 Part II, Pacific Northwest National Laboratory, 2018.

[26] P. Pullonen *et al.*, "Actively secure two-party computation: Efficient beaver triple generation," *Instructor*, 2013.

[27] G. M. Fichtenholz, *A Course of Differential and Integral Calculus (Chinese Translation)*. Beijing, China: Higher Education Press, 2006.

[28] D. P. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," in *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, pp. 1–5, 2008.

[29] "Weather underground: Weather record for columbus." `https://www.wunderground.com/`.

[30] W. Marion and K. Urban, "Users manual for TMY2s: Derived from the 1961–1990 national solar radiation data base," tech. rep., National Renewable Energy Lab., Golden, CO (United States), 1995.

[31] J. Lian, H. Ren, Y. Sun, and D. J. Hammerstrom, "Performance evaluation for transactive energy systems using double-auction market," *IEEE Transactions on Power Systems*, vol. 34, pp. 4128–4137, September 2019.