

# Secure and Privacy-Preserving Consensus

Minghao Ruan, Huan Gao, *Student Member, IEEE*, and Yongqiang Wang, *Senior Member, IEEE*

**Abstract**—Consensus is fundamental for distributed systems since it underpins key functionalities of such systems ranging from distributed information fusion, decision-making, to decentralized control. In order to reach an agreement, existing consensus algorithms require each agent to exchange explicit state information with its neighbors. This leads to the disclosure of private state information, which is undesirable in cases where privacy is of concern. In this paper, we propose a novel approach for undirected networks which can enable secure and privacy-preserving average consensus in a decentralized architecture in the absence of an aggregator or third-party. By leveraging partial homomorphic cryptography to embed secrecy in pairwise interaction dynamics, our approach can guarantee convergence to the consensus value (subject to a quantization error) in a *deterministic* manner without disclosing a node's state to its neighbors. We provide a new privacy definition for dynamical systems, and give a new framework to rigorously prove that a node's privacy can be protected as long as it has at least one legitimate neighbor which follows the consensus protocol faithfully without attempts to infer other nodes' states. In addition to enabling resilience to passive attackers aiming to steal state information, the approach also allows easy incorporation of defending mechanisms against active attackers who try to alter the content of exchanged messages. Furthermore, in contrast to existing noise-injection based privacy-preserving mechanisms which have to reconfigure the entire network when the topology or number of nodes varies, our approach is applicable to dynamic environments with time-varying coupling topologies. This secure and privacy-preserving approach is also applicable to weighted average consensus as well as maximum/minimum consensus under a new update rule. Numerical simulations and comparison with existing approaches confirm the theoretical results. Experimental results on a Raspberry-Pi board based micro-controller network are also presented to verify the effectiveness and efficiency of the approach.

**Index Terms**—Consensus, privacy preservation, security, partial homomorphic cryptography, digital signature.

## I. INTRODUCTION

As a building block of distributed computing, average consensus as well as its various variants such as weighted average consensus and maximum/minimum consensus has been an active research topic in computer science and optimization for decades [2], [3]. In recent years, with the advances of wireless communications and embedded systems, particularly the advent of wireless sensor networks and the Internet-of-Things, average consensus is finding increased applications in fields as diverse as automatic control, signal processing, social sciences, robotics, and optimization [4].

Part of the results was accepted for presentation at the Cyber-Physical Systems Security and Privacy Workshop [1]. The work was supported in part by the National Science Foundation under Grant 1824014 and 1738902.

Minghao Ruan, Huan Gao, and Yongqiang Wang are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA. Corresponding author: Yongqiang Wang (yongqiw@clemson.edu).

Conventional consensus approaches employ the explicit exchange of state values among neighboring nodes to reach agreement on the computation. Such an explicit exchange of state information has two potential problems. First, it results in breaches of the privacy of participating nodes who may want to keep their state information confidential. For example, a group of individuals using average consensus to compute a common opinion may want to keep their personal opinions secret [5]. Another example is power systems where multiple generators want to reach agreement on cost while keeping their individual generation information private [6]. Secondly, storing or exchanging information in the unencrypted plaintext form is vulnerable to attackers who can steal information by hacking into the communication links or even the nodes. With the increased number of reported attack events and the growing awareness of security, keeping data encrypted in storage and communications has become the norm in many applications, particularly in many real-time sensing and control systems such as power systems and wireless sensor networks.

To address the pressing need for privacy and security in consensus, recently several solutions have been proposed. Most existing approaches use the idea of obfuscation to mask the true state values by adding noise on the state. Motivated by database privacy in computer science, [7]–[11] exploited differential privacy to inject uncorrelated noise in average consensus. However, differential-privacy based approaches do not provide the exact average value due to their fundamental trade-off between enabled privacy and computational accuracy. To guarantee computational accuracy, [12]–[14] proposed to inject correlated noise to exchanged messages to guarantee convergence to the exact value, which, however, also leads to vulnerabilities to external eavesdroppers, as shown in our numerical simulations in Fig. 8 and Fig. 9. Furthermore, these approaches normally rely on the assumption of a *time-invariant* interaction graph, which is difficult to satisfy in many practical applications where the interaction patterns may change due to node mobility or fading communication channels. Observability based approaches have also been discussed to protect the privacy of multi-agent networks. The basic idea is to design the interaction topology so as to minimize the observability from a compromised agent, which amounts to minimizing its ability to infer the initial states of other network agents [15]–[17]. However, these approaches cannot protect the privacy of the direct neighbors of the compromised agent.

Neither can the aforementioned approaches protect nodes from active attackers who try to steal or even alter exchanged information by hacking into the nodes or the communication channels. To improve resilience to such attacks, a common approach is to employ cryptography. However, it is worth noting that although cryptography based approaches can easily provide privacy and security when an aggregator or third-

party is available [18], like in cloud-based control or computation [19]–[21], their extension to completely *decentralized* average consensus *without any aggregators or third-parties* is extremely hard due to the difficulties in decentralized key management.

In this paper, we propose a homomorphic cryptography based approach for undirected networks that can guarantee the security and privacy of a node as long as it has at least one legitimate neighbor. Here, a legitimate neighbor is defined as a neighboring node who follows the consensus protocol faithfully without attempts to infer other nodes' states. Compared with differential-privacy based approaches in [7]–[11] which trade accuracy for privacy and correlated-noise based approaches in [12]–[14] which are vulnerable to external eavesdroppers, our approach can guarantee both the accuracy of consensus and resilience against external eavesdroppers. Despite the increased computational complexity caused by encryption, our approach is still manageable for resource constrained low-cost micro-controllers, as analyzed in detail in Sec. III and experimentally validated in Sec. VIII-C. Unlike the existing cryptography based average consensus approach in [22] and [23], our approach allows every participating node to access the exact final value, and hence can be used in numerous applications where each agent has to use the precise consensus value to achieve cooperative control or information fusion.

The main contributions of the paper are as follows: 1) To our knowledge, our paper is the first to unify encryption and control dynamics in a completely decentralized manner without the assistance of a third party; 2) Our paper provides a new privacy definition for dynamical systems, and gives a new framework to rigorously prove that a node's privacy can be protected as long as it has at least one legitimate neighbor; 3) Our approach can be easily extended to provide security against active attackers aiming to alter the content of exchanged messages; and 4) We experimentally verified the efficiency of our approach using a Raspberry-Pi board based micro-controller network.

The outline of this paper is as follows. Sec. II reviews the average consensus problem and the homomorphic cryptography, particularly the Paillier cryptosystem. Our confidential interaction protocol is introduced in Sec. III. In Sec. IV we theoretically analyze the condition and rate of convergence, followed by a systematic discussion of privacy guarantees as well as security enforcement mechanisms in Sec. V. We further demonstrate in Sec. VI that our confidential interaction protocol can be extended to weighted average consensus and maximum/minimum consensus under a new update rule. Some implementation issues are discussed in Sec. VII. Both numerical examples and hardware experiments (on Raspberry Pi boards) are presented in Sec. VIII. The conclusion is drawn in Sec. IX.

## II. BACKGROUND

In this section we briefly review the average consensus problem and the homomorphic encryption.

### A. Average Consensus

We follow the same convention as in [4] where a network of  $N$  agents is represented by an undirected graph  $G = (V, E, \mathbf{A})$  with node set  $V = \{v_1, v_2, \dots, v_N\}$ , edge set  $E \subset V \times V$ , and a weighted adjacency matrix  $\mathbf{A} = [a_{ij}]$  which satisfies  $a_{ij} > 0$  if  $(v_i, v_j) \in E$  and 0 otherwise. The set of neighbors of a node  $v_i$  is denoted as

$$N_i = \{v_j \in V | (v_i, v_j) \in E\} \quad (1)$$

Throughout this paper we assume that the graph is undirected and connected. Therefore,  $\mathbf{A}$  is symmetric

$$a_{ij}^{(k)} = a_{ji}^{(k)} > 0 \quad \forall (v_i, v_j) \in E \quad (2)$$

Note that the superscript  $k$  denotes that the weights are time-varying. Sometimes we drop  $k$  for the sake of notational simplicity, but it is worth noting that all discussions in the paper are always applicable under time-varying weights. To achieve average consensus, namely converging of all states  $x_i[k]$  ( $i = 1, 2, \dots, N$ ) to the average of initial values, i.e.,  $\frac{\sum_{j=1}^N x_j[0]}{N}$ , one commonly-used update rule is

$$x_i[k+1] = x_i[k] + \varepsilon \sum_{v_j \in N_i} a_{ij}^{(k)} (x_j[k] - x_i[k]) \quad (3)$$

where  $\varepsilon$  is a constant step size.

### B. Homomorphic Encryption

Our method to protect privacy and security is to encrypt the state. To this end, we briefly introduce a cryptosystem, more specifically the public-key cryptosystem which is applicable in open and dynamic networks without the assist of any trusted third party for key management. A public-key cryptosystem uses two keys – a private key (also called secret key) and a public key distributed publicly. Any person can encrypt messages using a public key, but such messages can only be decrypted by the agents who have access to the private key. Most popular cryptosystems such as RSA [24], ElGamal [25], and Paillier [26] are public-key cryptosystems. In this paper we focus on the Paillier cryptosystem. We adopt a simplified version of the Paillier cryptosystem from [27] which has the following basic functions:

- Key generation:
  - 1) Choose two large prime numbers  $p \in \mathbb{Z}$  and  $q \in \mathbb{Z}$  of equal bit-length and compute  $n = pq$ .
  - 2) Let  $\lambda = \phi(n) = (p-1)(q-1)$  where  $\phi(\cdot)$  is the Euler's totient function.
  - 3) Let  $\mu = \phi(n)^{-1} \bmod n$  which is the modular multiplicative inverse of  $\phi(n)$ .
  - 4) The public key  $k_p$  is then  $n$ .
  - 5) The private key  $k_s$  is then  $(\lambda, \mu)$ .
- Encryption ( $c = \mathcal{E}(m) \in \mathbb{Z}_{n^2}^*$ ):
 

Recall the definitions of  $\mathbb{Z}_n = \{z | z \in \mathbb{Z}, 0 \leq z < n\}$  and  $\mathbb{Z}_n^* = \{z | z \in \mathbb{Z}, 0 \leq z < n, \gcd(z, n) = 1\}$  where  $\gcd(a, b)$  is the greatest common divisor of  $a$  and  $b$ .

  - 1) Choose a random  $r \in \mathbb{Z}_n^*$ .
  - 2) The ciphertext is given by  $c = (n+1)^m r^n \bmod n^2$ , where  $m \in \mathbb{Z}_n, c \in \mathbb{Z}_{n^2}^*$ .

- Decryption ( $m = \mathcal{D}(c) \in \mathbb{Z}_n$ ):

- 1) Define the integer division function  $L(u) = \frac{u-1}{n}$ .
- 2) The plaintext is  $m = L(c^\lambda \bmod n^2) \mu \bmod n$ .

A cryptosystem is homomorphic if it allows certain computations to be carried out on the encrypted ciphertext. The Paillier cryptosystem is additive homomorphic because the ciphertext of  $m_1 + m_2$ , i.e.,  $\mathcal{E}(m_1 + m_2)$ , can be obtained from  $\mathcal{E}(m_1)$  and  $\mathcal{E}(m_2)$  directly:

$$\begin{aligned} \mathcal{E}(m_1, r_1)\mathcal{E}(m_2, r_2) &= ((n+1)^{m_1}r_1^n)((n+1)^{m_2}r_2^n) \bmod n^2 \\ &= ((n+1)^{m_1+m_2}(r_1r_2)^n) \bmod n^2 \\ &= \mathcal{E}(m_1 + m_2, r_1r_2) \end{aligned} \quad (4)$$

The dependency on random numbers  $r_1$  and  $r_2$  is explicitly shown in (4), yet they play no role in the decryption. Therefore, the following shorthand notation will be used instead:

$$\mathcal{E}(m_1)\mathcal{E}(m_2) = \mathcal{E}(m_1 + m_2) \quad (5)$$

Moreover, if we multiply the same ciphertext  $k \in \mathbb{Z}^+$  times, we can obtain

$$\mathcal{E}(m)^k = \prod_{i=1}^k \mathcal{E}(m) = \mathcal{E}\left(\sum_{i=1}^k m\right) = \mathcal{E}(km) \quad (6)$$

Notice however, the Paillier cryptosystem is not multiplicative homomorphic because  $k$  in (6) is in the plaintext form. Furthermore, the existence of the random number  $r$  in Paillier cryptosystem gives it resistance to dictionary attacks [28] which infer a key to an encrypted message by systematically trying all possibilities, like exhausting all words in a dictionary. Moreover, since Paillier cryptography only works on numbers that can be represented by binary strings, we multiply a real-valued state by a large integer  $Q$  before converting it to a binary string so as to ensure small quantization errors. The details will be discussed in Sec. VII-A.

### III. CONFIDENTIAL INTERACTION PROTOCOL

In this section, we propose a completely decentralized, third-party free confidential interaction protocol that can guarantee average consensus while protecting the privacy of all participating nodes. Instead of adding noise to hide the states, our approach combines encryption with randomness in the system dynamics, i.e., the coupling weights  $a_{ij}^{(k)} \in \mathbb{R}$ , to prevent two communicating parties in a pairwise interaction from exposing information to each other. In this way the states are free from being contaminated by covering noise, guaranteeing a deterministic convergence to the average (subject to a quantization error). The computational complexity of this protocol is also discussed in this section.

We first present details of our confidential interaction protocol based on (3). In particular we show how a node can obtain the weighted difference (7) between itself and any of its neighbor(s) without disclosing each other's state information:

$$\begin{aligned} \Delta x_{ij}[k] &= a_{ij}^{(k)}(x_j[k] - x_i[k]) \\ \Delta x_{ji}[k] &= a_{ji}^{(k)}(x_i[k] - x_j[k]) \\ \text{subject to } a_{ij}^{(k)} &= a_{ji}^{(k)} > 0 \end{aligned} \quad (7)$$

Plugging the state difference (7) into (3) gives a new formulation of average consensus

$$x_i[k+1] = x_i[k] + \varepsilon \sum_{v_j \in N_i} \Delta x_{ij}[k] \quad (8)$$

Notice that in a decentralized system it is impossible to protect the privacy of both nodes in a pairwise interaction if the protocol (7) is used without a third party distributing secret  $a_{ij}^{(k)}$ . This is due to the fact that even if we encrypt all the intermediate steps, if one node, for instance  $v_i$ , has access to  $a_{ij}^{(k)}$ , it can still infer the value of  $x_j[k]$  through  $x_j[k] = \Delta x_{ij}[k]/a_{ij}^{(k)} + x_i[k]$ . From now on, for the sake of simplicity in bookkeeping, we omit the superscript  $k$  in  $a_{ij}^{(k)}$ . But it is worth noting that all the results hold for time-varying weights.

We solve this problem by constructing each weight  $a_{ij}$  as the product of two random numbers, namely  $a_{ij} = a_{ji} = a_{i \rightarrow j}a_{j \rightarrow i}$  with  $a_{i \rightarrow j}$  randomly generated by and only known to node  $v_i$  and  $a_{j \rightarrow i}$  randomly generated by and only known to node  $v_j$ . We will show later that this weight construction approach renders two interacting nodes unable to infer each other's state while guaranteeing convergence to the average. Next, without loss of generality, we consider a pair of connected nodes  $(v_1, v_2)$  to illustrate the idea (cf. Fig. 1). For simplicity, we assume that the states  $x_1$  and  $x_2$  are scalar. Each node maintains its own public and private key pairs  $(k_{pi}, k_{si})$ ,  $i \in \{1, 2\}$ .

Due to symmetry, we only show how node  $v_1$  obtains the weighted state difference, i.e., the flow  $v_1 \rightarrow v_2 \rightarrow v_1$ . Before starting the information exchange, node  $v_1$  (resp.  $v_2$ ) generates its new random number  $a_{1 \rightarrow 2}$  (resp.  $a_{2 \rightarrow 1}$ ). First, node  $v_1$  sends its encrypted negative state  $\mathcal{E}_1(-x_1)$  as well as the public key  $k_{p1}$  to node  $v_2$ . Note that here the subscript in  $\mathcal{E}_1$  denotes encryption using the public key of node  $v_1$ . Node  $v_2$  then computes the encrypted  $a_{2 \rightarrow 1}$ -weighted difference  $\mathcal{E}_1(a_{2 \rightarrow 1}(x_2 - x_1))$  following the three steps below:

- 1) Encrypt  $x_2$  with  $v_1$ 's public key  $k_{p1}$ :  $x_2 \rightarrow \mathcal{E}_1(x_2)$ .
- 2) Compute the difference directly in ciphertext:

$$\mathcal{E}_1(x_2 - x_1) = \mathcal{E}_1(x_2 + (-x_1)) = \mathcal{E}_1(x_2)\mathcal{E}_1(-x_1) \quad (9)$$

- 3) Compute the  $a_{2 \rightarrow 1}$ -weighted difference in ciphertext:

$$\mathcal{E}_1(a_{2 \rightarrow 1}(x_2 - x_1)) = (\mathcal{E}_1(x_2 - x_1))^{a_{2 \rightarrow 1}} \quad (10)$$

Then  $v_2$  returns  $\mathcal{E}_1(a_{2 \rightarrow 1}(x_2 - x_1))$  to  $v_1$ . After receiving  $\mathcal{E}_1(a_{2 \rightarrow 1}(x_2 - x_1))$ ,  $v_1$  decrypts it using the private key  $k_{s1}$  and multiplies the result with  $a_{1 \rightarrow 2}$  to get the weighted difference  $\Delta x_{12}$ :

$$\begin{aligned} \mathcal{E}_1(a_{2 \rightarrow 1}(x_2 - x_1)) &\xrightarrow{\mathcal{D}_1} a_{2 \rightarrow 1}(x_2 - x_1) \\ \Delta x_{12} &= a_{1 \rightarrow 2}a_{2 \rightarrow 1}(x_2 - x_1) \end{aligned} \quad (11)$$

In a similar manner, the exchange  $v_2 \rightarrow v_1 \rightarrow v_2$  produces  $\mathcal{E}_2(a_{1 \rightarrow 2}(x_1 - x_2))$  for  $v_2$  who then decrypts the message and multiplies the result by its own multiplier  $a_{2 \rightarrow 1}$  to get  $\Delta x_{21}$

$$\begin{aligned} \mathcal{E}_2(a_{1 \rightarrow 2}(x_1 - x_2)) &\xrightarrow{\mathcal{D}_2} a_{1 \rightarrow 2}(x_1 - x_2) \\ \Delta x_{21} &= a_{2 \rightarrow 1}a_{1 \rightarrow 2}(x_1 - x_2) \end{aligned} \quad (12)$$

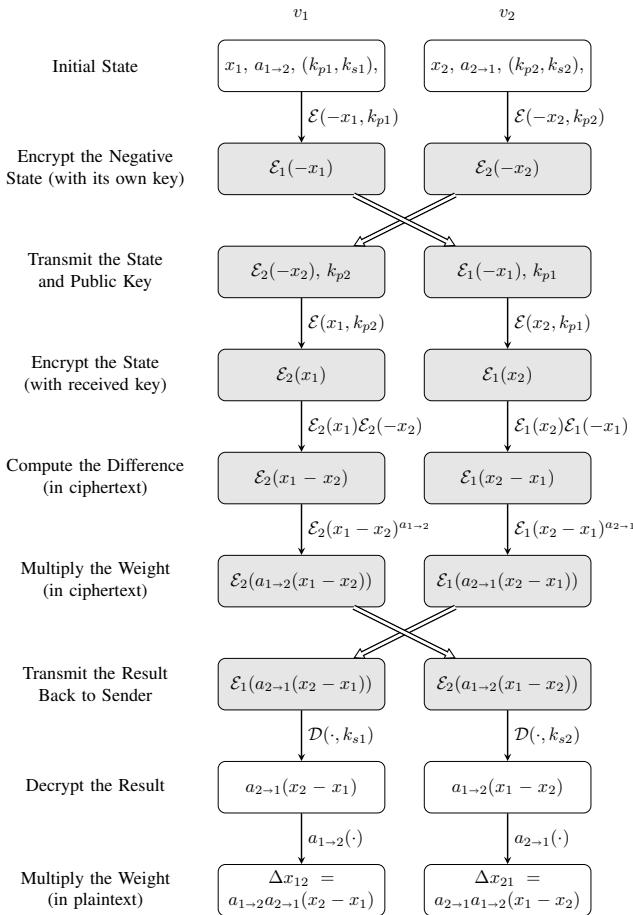


Fig. 1. A step-by-step illustration of the confidential interaction protocol. Single arrows indicate the flow of computations; double arrows indicate data exchange via a communication channel. Shaded nodes indicate the computation done in ciphertext. Note that  $a_{1 \rightarrow 2}$  and  $a_{2 \rightarrow 1}$  are different from step to step.

After each node collects the weighted differences from all neighbors, it updates its state with (8) accordingly.

Several remarks are in order:

- The construction of each  $a_{ij}$  as the product of two random numbers  $a_{i \rightarrow j}$  and  $a_{j \rightarrow i}$  is key to guarantee that the weights are symmetric, i.e.,  $a_{ij} = a_{ji}$ , which is necessary for average consensus. Note that the coupling weights  $a_{i \rightarrow j}$  are randomly chosen by individual nodes and are completely independent of communication channels.
- In the initial iteration  $k = 0$ , node  $v_i \in V$  chooses  $a_{i \rightarrow j}$  randomly from the set of real numbers  $\mathbb{R}$ , which is critical for privacy preservation, as will be clear in the proof of Theorem 3. After the initial iteration, i.e., for  $k \geq 1$ , node  $v_i \in V$  chooses  $a_{i \rightarrow j}$  randomly from a finite admissible range  $[\underline{a}, \bar{a}]$ . Note that  $a_{i \rightarrow j}$  can have any random distribution in  $[\underline{a}, \bar{a}]$  such as uniform, reciprocal, and truncated Gaussian.
- To guarantee the convergence of average consensus, the admissible range  $[\underline{a}, \bar{a}]$  must satisfy  $0 < \underline{a} < \bar{a} < \frac{1}{\sqrt{\varepsilon} \Delta}$  where  $\varepsilon$  is the constant step size in (3) and  $\Delta \triangleq \max_i |N_i|$  with  $|\bullet|$  denoting the set cardinality. Therefore, a smaller  $\varepsilon$  means a larger  $\bar{a}$ , which results in an extended

admissible range  $[\underline{a}, \bar{a}]$  in the selection of the random number  $a_{i \rightarrow j}$ .

- $v_2$  does not have the private key of  $v_1$  and cannot see  $x_1$  which is encrypted in  $E_1(-x_1)$ .
- Given  $a_{2 \rightarrow 1}(x_2 - x_1)$ ,  $v_1$  cannot solve for  $x_2$  because  $a_{2 \rightarrow 1}$  is only known to  $v_2$ .
- At each iteration, real-valued states are converted to fixed point representation for encryption; the weighted differences are converted back to real values for update.
- We encrypt  $E_1(-x_1)$  because it is much more difficult to compute subtraction in ciphertext. The issue regarding encrypting negative values using Paillier is discussed in Sec. VII.
- Different from [22] and [23] which deprive individual nodes from accessing the average value (note that in [23] individual participating nodes do not have the decryption key to decrypt the final consensus value which is in the encrypted form, otherwise they will be able to decrypt intermediate computations to access other nodes' states), our approach enables every node to calculate the average value in a completely decentralized manner, and hence can be used in numerous applications where each agent relies on the precise consensus value to achieve cooperative control or information fusion.

Next we analyze the complexity of our confidential interaction protocol. Note that, although the key generation process is typically the most time consuming step, it incurs a one-time fixed cost. Unless otherwise stated, we assume that the keys are reused for subsequent steps.

*Lemma 1:* Under our confidential interaction protocol illustrated in Fig. 1, for a particular node  $v_i$  in a connected network, the total computation overhead of each iteration is  $\mathcal{O}(|N_i|l)$  where  $|N_i|$  is the number of neighboring nodes of node  $v_i$  and  $l$  is the bit length of the public key  $n$ .

*Proof:* The encryption step requires two modulo exponentiation steps which can be computed in  $\mathcal{O}(l + \log(m))$  time, where  $m$  is the message's integer value and  $l$  is the bit length of the public key  $n$  [29]. The multiplication and modulo steps are both  $\mathcal{O}(1)$ . In practice the length of the public key  $n$  is much longer than the size of  $m$ , so the complexity of the encryption is assumed to be  $\mathcal{O}(l)$ .

The decryption step requires one modulo exponentiation, one subtraction, one division followed by a multiplication and a modulo operation. The overall time complexity is  $\mathcal{O}(l)$ .

During each message exchange, node  $v_i$  must encrypt its own message once and decrypt  $|N_i|$  return messages, so the overall complexity is  $\mathcal{O}((|N_i| + 1)l)$ . For each  $|N_i|$  in-bound messages, node  $v_i$  has to encrypt its own message, compute the difference, and then modulo-exponentiate the difference. Therefore the total computation complexity is  $\mathcal{O}(|N_i|(l + \log(\bar{a})))$ , where  $\bar{a}$  is the upper bound of the admissible range for the random selection of  $a_{i \rightarrow j}^{(k)}$ .

As a result, for a particular node  $v_i$ , the total computation overhead of each iteration is  $\mathcal{O}(|N_i|l)$ . ■

From Lemma 1, it can be seen that the computational complexity of our approach does not increase with network size but rather with the number of neighbors (so the computational complexity of our approach is moderate even for large

networks with moderate connections). In Sec. VIII-C, we will use a network of Raspberry Pi boards to experimentally verify that the computational complexity is easily manageable on resource-constrained micro-controller based real-time control systems.

#### IV. THEORETICAL ANALYSIS OF CONVERGENCE

In this section, we first show that average consensus will be guaranteed under the confidential interaction protocol. Then we provide convergence speed analysis of our approach.

Let  $\mathbf{x} \in \mathbb{R}^n$  denote the augmented state vector of all nodes. The network dynamics in (3) can be rewritten as:

$$\mathbf{x}[k+1] = \mathbf{P}^{(k)}\mathbf{x}[k] \quad (13)$$

where  $\mathbf{P}^{(k)} = \mathbf{I} - \varepsilon \mathbf{L}^{(k)}$  is the Perron matrix and  $\mathbf{L}^{(k)} = [l_{ij}^{(k)}]$  is the time-varying Laplacian matrix defined by

$$l_{ij}^{(k)} = \begin{cases} \sum_{j \in N_i} a_{ij}^{(k)} & i = j \\ -a_{ij}^{(k)} & i \neq j \end{cases} \quad (14)$$

*Theorem 1:* For a connected network of  $N$  nodes, if the coupling weights  $a_{ij}^{(k)}$  in (14) are established according to the confidential interaction protocol illustrated in Fig. 1 and the admissible range  $[\underline{a}, \bar{a}]$  for the random selection of  $a_{i \rightarrow j}^{(k)}$  satisfies  $0 < \underline{a} < \bar{a} < \frac{1}{\sqrt{\varepsilon \Delta}}$ , then the system will achieve average consensus with states converging to

$$\lim_{k \rightarrow \infty} \mathbf{x}[k] = \alpha \mathbf{1} \quad \text{with } \alpha = \text{Avg}[0] = \frac{1}{N} \mathbf{1}^T \mathbf{x}[0] \quad (15)$$

*Proof:* The proof can be obtained by following the similar line of reasoning of Theorem 2 in [4].  $\blacksquare$

*Remark 1:* Since the framework allows time-varying weighted matrix  $\mathbf{A}^{(k)}$ , it can easily be extended to the case with switching interaction graphs according to [30].

To analyze the convergence speed of our approach, following the convergence definition in [31] and [32], we define the convergence rate as at least  $\gamma \in (0, 1)$  if there exists a positive constant  $C$  such that  $\|\mathbf{x}[k] - \alpha \mathbf{1}\| \leq C\gamma^k$  holds for all  $k$ , where  $\alpha$  is the average value of the initial states of all nodes in (15). Note that a smaller  $\gamma$  means a faster convergence speed.

*Theorem 2:* For a connected network of  $N$  nodes, if the coupling weights  $a_{ij}^{(k)}$  in (14) are established according to the confidential interaction protocol illustrated in Fig. 1 and the admissible range  $[\underline{a}, \bar{a}]$  for the random selection of  $a_{i \rightarrow j}^{(k)}$  satisfies  $0 < \underline{a} < \bar{a} < \frac{1}{\sqrt{\varepsilon \Delta}}$ , then the convergence rate of our privacy-preserving average consensus algorithm is at least  $\gamma = (1 - \eta^{N-1})^{\frac{1}{N-1}} \in (0, 1)$  with  $\eta = \min\{1 - \varepsilon \Delta \bar{a}^2, \varepsilon \underline{a}^2\}$ , meaning that there exists a positive constant  $C$  such that  $\|\mathbf{x}[k] - \alpha \mathbf{1}\| \leq C\gamma^k$  holds for all  $k$ .

*Proof:* For iteration  $k = 0$ , since  $\mathbf{1}^T \mathbf{P}^{(0)} = \mathbf{1}^T$  holds, we have  $\mathbf{1}^T \mathbf{x}[1] = \mathbf{1}^T \mathbf{P}^{(0)} \mathbf{x}[0] = \mathbf{1}^T \mathbf{x}[0]$ . So we have  $\alpha = \frac{1}{N} \sum_{j=1}^N x_j[0] = \frac{1}{N} \sum_{j=1}^N x_j[1]$ .

For iteration  $k \geq 1$ , each random number  $a_{i \rightarrow j}^{(k)}$  is randomly chosen from  $[\underline{a}, \bar{a}]$  by node  $i$  with  $0 < \underline{a} < \bar{a} < \frac{1}{\sqrt{\varepsilon \Delta}}$ . Further taking into account  $\mathbf{P}^{(k)} = \mathbf{I} - \varepsilon \mathbf{L}^{(k)}$ , we have the following

properties for  $\mathbf{P}^{(k)}$ 's  $(i, i)$ th element  $p_{ii}^{(k)}$  and  $(i, j)$ th element  $p_{ij}^{(k)}$  for any  $1 \leq i, j \leq N$ :

$$p_{ii}^{(k)} = 1 - \varepsilon \sum_{j \in N_i} a_{i \rightarrow j}^{(k)} a_{j \rightarrow i}^{(k)} \geq 1 - \varepsilon \Delta \bar{a}^2 \quad \forall v_i \in V \quad (16)$$

and

$$\begin{cases} p_{ij}^{(k)} = 0 & \forall v_i \in V \text{ and } v_j \notin N_i \cup \{v_i\} \\ p_{ij}^{(k)} = \varepsilon a_{i \rightarrow j}^{(k)} a_{j \rightarrow i}^{(k)} \geq \varepsilon \underline{a}^2 & \forall v_i \in V \text{ and } v_j \in N_i \end{cases} \quad (17)$$

Defining  $\eta$  as  $\eta \triangleq \min\{1 - \varepsilon \Delta \bar{a}^2, \varepsilon \underline{a}^2\}$ , we have  $0 < \eta < 1$  due to  $0 < \underline{a} < \bar{a} < \frac{1}{\sqrt{\varepsilon \Delta}}$ . According to (16) and (17), we have  $p_{ij}^{(k)} \geq \eta$  for  $v_i \in V$  and  $v_j \in N_i \cup \{v_i\}$ .

Denote  $\mathbf{P}^{(k)} \mathbf{P}^{(k-1)} \dots \mathbf{P}^{(s)}$  as  $\Phi(k, s)$ . Note that  $\Phi(k, k) = \mathbf{P}^{(k)}$  holds for all  $k$ . Since  $\mathbf{P}^{(k)}$  ( $k \geq 1$ ) satisfies the Assumptions 2, 3, 4, and 5 in [33], following its Proposition 1, we have

$$\left| \phi_{ij}^{(k, s)} - \frac{1}{N} \right| \leq 2 \frac{1 + \eta^{-N+1}}{1 - \eta^{N-1}} (1 - \eta^{N-1})^{\frac{k-s}{N-1}} \quad (18)$$

for  $i, j = 1, 2, \dots, N$  and  $k \geq s \geq 1$  where  $\phi_{ij}^{(k, s)}$  represents the  $(i, j)$ th element of  $\Phi(k, s)$ .

From (13), we have

$$\mathbf{x}[k] = \mathbf{P}^{(k-1)} \mathbf{P}^{(k-2)} \dots \mathbf{P}^{(1)} \mathbf{x}[1] = \Phi(k-1, 1) \mathbf{x}[1] \quad (19)$$

for  $k \geq 2$ . Combining (16) and (17) with (19), we obtain

$$\begin{aligned} \|\mathbf{x}[k] - \alpha \mathbf{1}\| &= \|\Phi(k-1, 1) \mathbf{x}[1] - \alpha \mathbf{1}\| \\ &= \left( \sum_{i=1}^N \left| \sum_{j=1}^N (\phi_{ij}^{(k-1, 1)} - \frac{1}{N}) x_j[1] \right|^2 \right)^{1/2} \\ &\leq \left( \sum_{i=1}^N \left( \sum_{j=1}^N \left| \phi_{ij}^{(k-1, 1)} - \frac{1}{N} \right|^2 \|\mathbf{x}[1]\|^2 \right) \right)^{1/2} \\ &\leq 2N \frac{1 + \eta^{-N+1}}{1 - \eta^{N-1}} (1 - \eta^{N-1})^{\frac{k-2}{N-1}} \|\mathbf{x}[1]\| \end{aligned} \quad (20)$$

for  $k \geq 2$ , where we used Cauchy-Schwarz inequality in the derivation. By re-arranging the terms, we have

$$\|\mathbf{x}[k] - \alpha \mathbf{1}\| \leq C_0 \gamma^k \quad (21)$$

for  $k \geq 2$ , where

$$C_0 = 2N \|\mathbf{x}[1]\| \frac{1 + \eta^{-N+1}}{(1 - \eta^{N-1})^{\frac{N+1}{N-1}}} \quad (22)$$

and

$$\gamma = (1 - \eta^{N-1})^{\frac{1}{N-1}} \quad (23)$$

Given  $\eta \in (0, 1)$ , it can be easily verified that  $\gamma \in (0, 1)$  holds.

Defining  $C$  as

$$C \triangleq \max\{\|\mathbf{x}[0] - \alpha \mathbf{1}\|, \|\mathbf{x}[1] - \alpha \mathbf{1}\| \gamma^{-1}, C_0\}$$

we have

$$\|\mathbf{x}[k] - \alpha \mathbf{1}\| \leq C \gamma^k \quad (24)$$

for all  $k$ . Therefore, the convergence rate of our privacy-preserving average consensus algorithm is at least  $\gamma = (1 - \eta^{N-1})^{\frac{1}{N-1}} \in (0, 1)$ .  $\blacksquare$

From Theorem 2, it can be seen that the initial weights will affect the absolute convergence time as they affect the parameter  $C$  in (24). However, they will not affect the exponential convergence rate  $\gamma$  according to (23). According to (24), it is easy to see that a smaller  $\gamma$  leads to a faster convergence. Given the relationship  $\gamma = (1 - \eta^{N-1})^{\frac{1}{N-1}}$ , to get a faster convergence speed, i.e., a smaller  $\gamma$ , it suffices to increase  $\eta$ . Further noting  $\eta = \min\{1 - \varepsilon \Delta \bar{a}^2, \varepsilon \underline{a}^2\}$ , increasing  $\eta$  amounts to decreasing  $\bar{a}$  and/or increasing  $\underline{a}$ , which in turn reduces the size of the range  $[\underline{a}, \bar{a}]$ . Therefore, there is a trade-off between the size of the admissible range  $[\underline{a}, \bar{a}]$  and the convergence speed. However, this trade-off only matters if one wants to protect the intermediate states from being inferable by honest-but-curious attackers (a reduced  $[\underline{a}, \bar{a}]$  reduces the admissible range for randomizing  $a_{i \rightarrow j}$  and  $a_{j \rightarrow i}$ , and hence enables an honest-but-curious attacker  $v_i$  to get a better range estimation of its neighbor  $v_j$ 's intermediate states  $x_j[k]$  ( $k \geq 1$ ) based on received  $\Delta x_{ij}$ ). If the intermediate states do not need to be protected as in the consensus problem considered in this paper (where only the initial state values matter), we can still enhance convergence speed by reducing the size of the admissible range  $[\underline{a}, \bar{a}]$ . In fact, even if we reduce the size of  $[\underline{a}, \bar{a}]$  to zero for iterations after  $k = 0$ , i.e., setting  $\underline{a} = \bar{a}$  starting from  $k = 1$  (note that in the initial iteration  $k = 0$  coupling weights  $a_{i \rightarrow j}^{(0)}$  should be randomly chosen from  $\mathbb{R}$ ), our approach can still guarantee the privacy of initial state values, as is clear from the proof of Theorem 3 in Sec. V-A.

## V. ANALYSIS OF PRIVACY AND SECURITY

Privacy and security are sometimes used interchangeably in the literature but here we make the distinction explicit. Among the control community privacy is equivalent to the concept of unobservability. Privacy is also closely related to the concept of semantic security from cryptography [28]. Both concepts essentially concern with an honest-but-curious adversary which is interested in learning the states of the network but conforms to the rules of the system. Security, on the other hand, deals with a broader issue which includes learning the states as well as the possibilities of exploiting the system to cause damages.

### A. Privacy Guarantees

Our protocol provides protection against an honest-but-curious adversary, which can be a node in the network or an observer eavesdropping communication links.

The Paillier encryption algorithm is known to provide semantic security, i.e., Indistinguishability under Chosen Plaintext Attack (IND-CPA) [26]. As a result, the recipient of the first transmission  $\mathcal{E}(-x_i)$  cannot see the value of  $x_i$  at all time.

Before analyzing the privacy-preserving performance of our approach, we first give a definition of privacy preservation used throughout this paper.

**Definition 1:** For a connected network of  $N$  nodes, the privacy of the initial value  $x_i[0]$  of node  $v_i$  is preserved if an honest-but-curious adversary cannot estimate the value of  $x_i[0]$  with any accuracy.

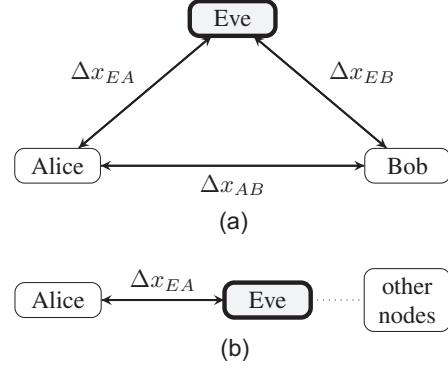


Fig. 2. Two connection configurations considered in the proofs of Theorem 3 (subplot a) and Theorem 4 (subplot b).

Definition 1 requires that an honest-but-curious adversary cannot even find a range for a private value and thus is more stringent than the privacy preservation definition considered in [34]–[36] which defines privacy preservation as the inability of an adversary to *uniquely* determine the protected value. It is worth noting that by a finite range, we mean lower and upper bounds that an adversary may obtain based on accessible information. We do not consider representation bounds caused by the finite number of bytes that can be used to represent a number in a computer.

As per the naming convention in cryptography, it is customary to name the legitimate sender and receiver participants as  $A$  (Alice) and  $B$  (Bob), and the adversary as  $E$  (Eve).

**Theorem 3:** For a connected network of  $N$  nodes with system dynamics in (3), we assume that all nodes follow the confidential interaction protocol illustrated in Fig. 1. An honest-but-curious node Eve who can receive messages from a neighboring node Alice cannot learn the initial state of Alice if Alice is also connected to another legitimate node Bob.

**Proof:** Without loss of generality, we consider the connection configuration illustrated in Fig. 2(a) where Eve can interact with both Alice and Bob. If Eve cannot infer the state of Alice or Bob in this configuration, neither can it when either the Alice–Eve connection or the Bob–Eve connection is removed which reduces the amount of information accessible to Eve.

We propose a new privacy-proving approach based on the indistinguishability of a private value's arbitrary variation to Eve. We define the information accessible to Eve at iteration  $k$  as  $I_E[k] = \{\Delta x_{EA}[k], \Delta x_{EB}[k], x_E[k], a_{E \rightarrow A}^{(k)}, a_{E \rightarrow B}^{(k)}\}$ . So as time evolves, the cumulated information accessible to Eve can be summarized as  $I_E = \bigcup_{k=0}^{\infty} I_E[k]$ .

To show that the privacy of the initial value  $x_A[0]$  can be preserved against Eve, i.e., Eve cannot estimate the value of  $x_A[0]$  with any accuracy, it suffices to show that under any initial value  $\bar{x}_A[0] \neq x_A[0]$  the information accessible to Eve, i.e.,  $\bar{I}_E$  could be exactly the same as  $I_E$ , the cumulated information accessible to Eve under  $x_A[0]$ . This is because the only information available for Eve to infer the initial value  $x_A[0]$  is  $I_E$ , and if  $I_E$  could be the outcome under any initial values of  $x_A[0]$ , then Eve has no way to even find a range for

the initial value  $x_A[0]$ . Therefore, we only need to prove that for any  $\bar{x}_A[0] \neq x_A[0]$ ,  $\bar{I}_E = I_E$  could hold.

Next we show that there exist initial values of  $x_B[0]$  and coupling weights satisfying the requirements of the confidential interaction protocol in Sec. III that make  $\bar{I}_E = I_E$  hold under  $\bar{x}_A[0] \neq x_A[0]$ . (Note that the alternative initial values of  $x_B[0]$  should guarantee that the agents still converge to the original average value after  $x_A[0]$  is altered to  $\bar{x}_A[0]$ .) More specifically, under the following initial condition

$$\bar{x}_B[0] = x_A[0] + x_B[0] - \bar{x}_A[0] \quad (25)$$

and coupling weights

$$\begin{aligned} \bar{a}_{E \rightarrow A}^{(0)} &= a_{E \rightarrow A}^{(0)} \\ \bar{a}_{E \rightarrow B}^{(0)} &= a_{E \rightarrow B}^{(0)} \\ \bar{a}_{A \rightarrow E}^{(0)} &= a_{A \rightarrow E}^{(0)}(x_E[0] - x_A[0])/(x_E[0] - \bar{x}_A[0]) \\ \bar{a}_{A \rightarrow B}^{(0)} &= (x_A[0] - \bar{x}_A[0])/(\varepsilon a_{A \rightarrow B}^{(0)}(x_B[0] - x_A[0])) + a_{A \rightarrow B}^{(0)} \\ \bar{a}_{B \rightarrow E}^{(0)} &= a_{B \rightarrow E}^{(0)}(x_E[0] - x_B[0])/(x_E[0] - \bar{x}_B[0]) \\ \bar{a}_{B \rightarrow A}^{(0)} &= a_{B \rightarrow A}^{(0)}(x_B[0] - x_A[0])/(\bar{x}_B[0] - \bar{x}_A[0]) \\ \bar{a}_{i \rightarrow j}^{(k)} &= a_{i \rightarrow j}^{(k)} \quad \forall i, j \in \{A, B, E\}, k = 1, 2, \dots \end{aligned} \quad (26)$$

it can be easily verified that  $\bar{I}_E = I_E$  holds for any  $\bar{x}_A[0] \neq x_A[0]$ . Note that (25) is used to guarantee that the consensus value does not change under the new initial values  $\bar{x}_A[0]$  and  $\bar{x}_B[0]$ . Therefore, the honest-but-curious node Eve cannot learn the initial state of a neighboring node Alice based on accessible information if Alice is also connected to another legitimate node Bob.  $\blacksquare$

It is worth noting that conventional observability based approaches in e.g., [16] and [17] cannot be used to analyze node Eve's ability to infer the states of Alice because the coupling weights  $a_{A \rightarrow E}^{(k)}$  are not available to Eve.

*Remark 2:* In the confidential interaction protocol, we allow the coupling weights for  $k = 0$  to be unrestricted and random chosen from  $\mathbb{R}$ . This unrestricted admissible range for coupling weights of  $k = 0$  is key to guarantee that no matter what value  $\bar{x}_A[0]$  is, the resulting  $\bar{a}_{E \rightarrow A}^{(0)}$ ,  $\bar{a}_{E \rightarrow B}^{(0)}$ ,  $\bar{a}_{A \rightarrow E}^{(0)}$ ,  $\bar{a}_{A \rightarrow B}^{(0)}$ ,  $\bar{a}_{B \rightarrow E}^{(0)}$ , and  $\bar{a}_{B \rightarrow A}^{(0)}$  in the left-hand-side of (26) are always admissible and hence guarantee the achievement of the defined privacy.

*Remark 3:* From the derivation in Theorem 3, one can get that if only initial states need to be protected, as defined in Definition 1, then encrypting messages for the initial iteration  $k = 0$  is sufficient. Avoiding encrypting messages after the initial iteration can greatly reduce the computational complexity to a one-time cost and is highly desirable when computational complexity is of concern. Of course, if all intermediate messages need to be protected from being disclosed to an external eavesdropper, we still need to encrypt exchanged messages after the initial iteration.

*Remark 4:* Given that the random distribution of  $a_{i \rightarrow j}$  can be non-stationary, the approach is also resilient to statistical inferences based on law-of-large-numbers [37].

*Remark 5:* From the derivation, we can see that when Alice is connected to at least one legitimate node Bob, the collusion of multiple honest-but-curious nodes does not help the inference of Alice's initial state.

*Remark 6:* Following the same line of reasoning, it can be obtained that an honest-but-curious node Eve 1 cannot infer the initial state of a neighboring node Alice if Alice is also connected to another honest-but-curious node Eve 2 that does not collude with Eve 1.

Based on the analysis framework, we can also obtain a situation where it is possible for Eve to infer other nodes' states which should be avoided.

*Theorem 4:* For a connected network of  $N$  nodes with system dynamics in (3), we assume that all nodes follow the confidential interaction protocol illustrated in Fig. 1. If a node Alice is connected to the rest of the network only through an (or a group of colluding) honest-but-curious node(s) Eve, Alice's initial state can be uniquely inferred by Eve in an asymptotic sense.

*Proof:* If Alice is directly connected to multiple honest-but-curious nodes that collude with each other, then these nodes can share information with each other to cooperatively estimate Alice's initial state, and hence can be regarded as one node. Therefore, we can only consider the case in which Alice is only connected to one honest-but-curious node Eve, as illustrated in Fig. 2(b). In this case, from the perspective of the honest-but-curious node Eve, the measurement seen at each time step  $k$  is  $\Delta x_{EA}[k] = a_{A \rightarrow E}^{(k)} a_{E \rightarrow A}^{(k)} (x_A[k] - x_E[k])$  with  $\Delta x_{EA}[k] = -\Delta x_{AE}[k]$ . From the system dynamics in (8), Eve knows

$$\begin{aligned} x_A[0] &= x_A[P] - \varepsilon \sum_{k=0}^{P-1} \Delta x_{AE}[k] \\ &= x_A[P] + \varepsilon \sum_{k=0}^{P-1} \Delta x_{EA}[k] \end{aligned} \quad (27)$$

and hence can construct the following observer to estimate the initial value of  $x_A[0]$ :

$$\hat{x}_A[0] = x_E[P] + \varepsilon \sum_{k=0}^{P-1} \Delta x_{EA}[k] \quad (28)$$

As  $P$  goes to infinity, average consensus will be achieved asymptotically, i.e.,  $\lim_{P \rightarrow \infty} x_A[P] = \lim_{P \rightarrow \infty} x_E[P] = \alpha = \sum_{i=1}^N x_i[0]/N$ , which leads to

$$\begin{aligned} \lim_{P \rightarrow \infty} \hat{x}_A[0] &= \lim_{P \rightarrow \infty} (x_E[P] + \varepsilon \sum_{k=0}^{P-1} \Delta x_{EA}[k]) \\ &= \lim_{P \rightarrow \infty} (x_A[P] + \varepsilon \sum_{k=0}^{P-1} \Delta x_{EA}[k]) \\ &= x_A[0] \end{aligned} \quad (29)$$

Therefore Eve can uniquely infer Alice's initial state  $x_A[0]$  in an asymptotic sense through the estimator (28).  $\blacksquare$

Theorem 4 implies that the single neighbor configuration should always be avoided, which is also required by other noise-based privacy protocols such as [12] and [13].

## B. Security Solution

Due to the additive homomorphic property, the Paillier cryptosystem is vulnerable to active adversaries who may

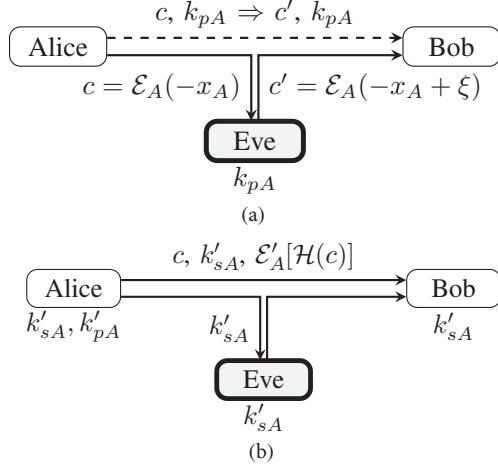


Fig. 3. Illustration of attacks from an active attacker (a) and the defense mechanism with a digital signature (b).

attempt to alter the message being sent through the channel. Although such adversaries cannot find out the exact states of the communicating nodes, they can still inflict significant damage to the system.

Consider the scenario where the communication from node Alice to Bob is intercepted by an active adversary Eve (cf. Fig. 3(a)). Since Alice's public key  $k_{pA}$  is sent along with  $\mathcal{E}(-x_1)$ , Eve may use the additive homomorphism to inject an arbitrary noise  $\xi$  to the original message  $\mathcal{E}(-x_1)$  to sway it to  $\mathcal{E}(-x_1 + \xi)$ . If Bob has no way to tell whether the received message has been modified, Eve may exploit this vulnerability to make the network either converge to a wrong value or not converge at all.

In applications where security is of prime concern, it is imperative to be able to verify the integrity of any received message. Here we propose a solution to provide resilience against active adversaries. Our basic idea is to attach a digital signature to exchanged messages in the confidential interaction protocol, as illustrated in Fig. 3(b). The signature requires an additional pair of public/private keys  $(k'_{pA}, k'_{sA})$  and a hash function  $\mathcal{H}(\cdot)$ , and is represented as  $(k'_{sA}, \mathcal{E}'_A[\mathcal{H}(c)])$  where  $c$  is the encrypted message. The additional private key  $k'_{sA}$  is sent so that Bob can decrypt  $\mathcal{E}'_A[\mathcal{H}(c)]$  and check if the resulting  $\mathcal{H}(c)$  matches the received  $c$  in terms of the hash operation  $\mathcal{H}(\cdot)$ . Based on the attached digital signature, the recipient can verify possible modifications during communication.

**Lemma 2:** When Alice and Bob use the mechanism in Fig. 3(b) for message transmission and integrity verification, Eve cannot fake Alice's identity by forging a signature that could pass the verification of Bob.

**Proof:** From the defense mechanism illustrated in Fig. 3(b), it can be seen that without the public key  $k'_{pA}$ , Eve cannot forge a valid signature (that can be decrypted by Bob), any Eve's attempt to modify  $c$  will cause a mismatch between received  $c$  and decrypted  $\mathcal{H}(c)$  in terms of the hash operation  $\mathcal{H}(\cdot)$  (cf. [38] for details). Therefore, the defense mechanism can provide resilience against active adversaries. ■

## VI. EXTENSIONS TO OTHER CONSENSUS

Using the same confidential interaction protocol, we can ensure the privacy of other variants of average consensus. Here we show the applications to three other commonly used consensus problems, i.e., the weighted average consensus, maximum consensus, and minimum consensus.

### A. Weighted Average Consensus

Weighted average consensus seeks convergence of all states to a weighed sum of the initial state, i.e.,  $\frac{\sum_{j=1}^N w_j x_j[0]}{\sum_{j=1}^N w_j}$  with  $w_j > 0$  being the weights. According to [4], weighted average consensus can be achieved by using the following update rule:

$$x_i[k+1] = x_i[k] + \frac{\varepsilon}{w_i} \sum_{v_j \in N_i} a_{ij}^{(k)} (x_j[k] - x_i[k]) \quad (30)$$

Note that the average consensus is in fact a special case of the weighted average consensus with all  $w_i$  being equal to the same constant.

**Theorem 5:** For a connected network of  $N$  nodes, if the coupling weights  $a_{ij}^{(k)}$  in (30) are established according to the confidential interaction protocol illustrated in Fig. 1 and the admissible range  $[a, \bar{a}]$  for the random selection of  $a_{i \rightarrow j}^{(k)}$  satisfies  $0 < a < \bar{a} < \sqrt{\frac{\min_i w_i}{\varepsilon \Delta}}$ , then the system will achieve weighted average consensus with states converging to

$$\lim_{k \rightarrow \infty} x_i[k] = \frac{\sum_{j=1}^N w_j x_j[0]}{\sum_{j=1}^N w_j}, \quad i = 1, 2, \dots, N. \quad (31)$$

*Proof:* The proof can be obtained by following a similar line of reasoning of Theorem 1 in [39]. ■

### B. Maximum Consensus

The maximum consensus seeks the convergence of all states to the maximal initial value among all states, i.e.,  $\lim_{k \rightarrow \infty} x_i[k] = \max_j x_j[0]$  for  $i = 1, 2, \dots, N$ . Next we show how to use the confidential interaction protocol in Sec. III to guarantee the privacy in maximum consensus. In contrast to the conventional maximum consensus protocol where the current state of a node is directly replaced with the maximum state among its neighbors [40], we formulate maximum consensus problem as a nonlinear dynamical system to fit into our confidential interaction protocol.

We design the update rule as

$$x_i[k+1] = x_i[k] + \max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) \quad (32)$$

Compared with the average consensus protocol, the difference is that we replaced the summation operation with the "max" operator. What is more important is the inclusion of the node itself in the computation, which ensures that the output of the max operator is non-negative.

In this new maximum consensus approach, the weighted state difference can still be calculated using the confidential interaction protocol in Sec. III, but the operation of multiplying a node's own random weight to the weighted difference, i.e.,  $a_{1 \rightarrow 2}$  in (11), can be removed to simplify the computation because there is no need to guarantee a symmetric interaction

graph any more. For this reason, here we always set  $a_{1 \rightarrow 2}$  in (11) to 1, which leads to the coupling weight in (32) above. Furthermore, to guarantee the convergence of maximum consensus, the admissible range  $[\underline{a}, \bar{a}]$  for the random selection of  $a_{j \rightarrow i}^{(k)}$  should satisfy  $0 < \underline{a} < \bar{a} < 1$ .

**Theorem 6:** For a connected network of  $N$  nodes, under the confidential interaction protocol illustrated in Fig. 1 and  $a_{j \rightarrow i}^{(k)} \in [\underline{a}, \bar{a}]$  for  $k \geq 0$  with  $0 < \underline{a} < \bar{a} < 1$ , the update rule (32) can achieve maximum consensus, i.e.,

$$\lim_{k \rightarrow \infty} x_i[k] = \max_{1 \leq j \leq N} x_j[0], \quad i = 1, 2, \dots, N. \quad (33)$$

*Proof:* According to (32), we have

$$\begin{aligned} x_i[k+1] - x_i[k] &= \max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) \\ &\geq a_{i \rightarrow i}^{(k)} (x_i[k] - x_i[k]) = 0 \end{aligned} \quad (34)$$

So  $x_i[k+1] - x_i[k] \geq 0$  always holds for  $i = 1, 2, \dots, N$ , i.e., the value of each node is non-decreasing.

Let  $x_p[k] = \max_i x_i[k]$  at step  $k$ . According to (32) and  $0 < a_{j \rightarrow p}^{(k)} < 1$ , we have

$$x_p[k+1] - x_p[k] = \max_{v_j \in N_p \cup \{v_p\}} a_{j \rightarrow p}^{(k)} (x_j[k] - x_p[k]) \leq 0 \quad (35)$$

Since the value of each node is non-decreasing, we have  $x_p[k+1] = x_p[k]$ , i.e.,  $x_p[k]$  is time-invariant.

Next we show that all the other nodes will stay less or equal to  $x_p[k]$ . For every node  $i$ , we have

$$x_i[k+1] = x_i[k] + \max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) \quad (36)$$

Since  $\max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) \geq 0$  holds, we have

$$\begin{aligned} \max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) &\leq \max_{v_j \in N_i \cup \{v_i\}} (x_j[k] - x_i[k]) \\ &\leq \max_{v_j \in N_i} (x_j[k] - x_i[k]) \leq x_p[k] - x_i[k] \end{aligned} \quad (37)$$

where we used the facts  $0 < a_{j \rightarrow i}^{(k)} < 1$  and  $x_p[k] = \max_i x_i[k]$ . Combing (37) with (36) leads to

$$\begin{aligned} x_i[k+1] &= x_i[k] + \max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) \\ &\leq x_i[k] + x_p[k] - x_i[k] = x_p[k] \end{aligned} \quad (38)$$

As a result, the maximum value is invariant with respect to  $k$ .

Let  $\max_i x_i[k] = \max_i x_i[0] = \alpha$ . We can now define the error vector as  $\delta[k] = \alpha \mathbf{1} - \mathbf{x}[k]$ , where  $\mathbf{x}[k] = [x_1[k], x_2[k], \dots, x_N[k]]^T$ . Note that  $\delta_i[k] \geq 0$  holds for all  $i$  and  $k$ . Theorem 6 is thus equivalent to proving

$$\lim_{k \rightarrow \infty} \delta[k] = \mathbf{0} \quad (39)$$

According to (32), the dynamics of  $\delta_i[k]$  is given by

$$\delta_i[k+1] - \delta_i[k] = - \max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) \quad (40)$$

Define the Lyapunov function  $V(\delta[k]) = \delta[k]^T \delta[k] \geq 0$ , where the equality holds only when  $\delta[k] = \mathbf{0}$ , then we have

$$V(\delta[k+1]) - V(\delta[k]) = \delta[k+1]^T \delta[k+1] - \delta[k]^T \delta[k] \quad (41)$$

Expanding the right-hand-side (RHS) yields:

$$\begin{aligned} \text{RHS} &= (\delta[k+1] - \delta[k])^T (\delta[k+1] + \delta[k]) \\ &= - \sum_{v_i \in V} \max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) (\delta_i[k+1] + \delta_i[k]) \\ &\leq 0 \end{aligned} \quad (42)$$

The equality holds when  $\max_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) = 0$  is true for  $i = 1, 2, \dots, N$ , i.e., when the maximum consensus is achieved. So the Lyapunov function  $V(\delta[k])$  will keep decreasing until the maximum consensus is achieved. ■

### C. Minimum Consensus

In contrast to maximum consensus, the minimum consensus problem seeks the convergence of all states to the minimal initial value among all states, i.e.,  $\lim_{k \rightarrow \infty} x_i[k] = \min_j x_j[0]$  for  $i = 1, 2, \dots, N$ . Following the same idea as maximum consensus, we propose a new update rule for achieving minimum consensus to fit into our confidential interaction protocol:

$$x_i[k+1] = x_i[k] + \min_{v_j \in N_i \cup \{v_i\}} a_{j \rightarrow i}^{(k)} (x_j[k] - x_i[k]) \quad (43)$$

Similar to Theorem 6, we can obtain the following results for minimum consensus:

**Theorem 7:** For a connected network of  $N$  nodes, under the confidential interaction protocol illustrated in Fig. 1 and  $a_{j \rightarrow i}^{(k)} \in [\underline{a}, \bar{a}]$  for  $k \geq 0$  with  $0 < \underline{a} < \bar{a} < 1$ , the update rule (43) can achieve minimum consensus, i.e.,

$$\lim_{k \rightarrow \infty} x_i[k] = \min_{1 \leq j \leq N} x_j[0], \quad i = 1, 2, \dots, N. \quad (44)$$

*Proof:* The proof follows the same line of reasoning as Theorem 6 and is omitted. ■

## VII. IMPLEMENTATION DETAILS

In addition to the constraint imposed on  $a_{i \rightarrow j}^{(k)}$ , there are other technical issues that must be addressed for the implementation of our confidential interaction protocol.

### A. Quantization

Real-world applications typically have  $x_i \in \mathbb{R}$  which are represented by floating point numbers in modern computing architectures. On the contrary, encryption algorithms only work on unsigned integers. Define the casting function  $f(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \rightarrow \mathcal{M} \subset \mathbb{Z}$  and its inverse  $f^{-1}(\cdot, \cdot) : \mathcal{M} \times \mathbb{R} \rightarrow \mathbb{R}$  as

$$f(x, Q) = \lceil Qx \rceil_{\mathcal{M}}, \quad f^{-1}(y, Q) = \frac{y}{Q} \quad (45)$$

where  $\lceil \cdot \rceil_{\mathcal{M}}$  maps the input to the nearest integer in  $\mathcal{M}$ . For the Paillier cryptosystem, this mapping is equivalent to the rounding operation, hence the step size is 1 which is uniform. Consequently the maximum quantization error is bounded by

$$\max_{x \in \mathbb{R}} |x - f^{-1}(f(x, Q), Q)| = \frac{1}{Q} \quad (46)$$

Although the quantization error is unavoidable, it can be made arbitrarily small by using a large enough  $Q$  in (45). In practice we choose a sufficiently large value for  $Q$  so that

the quantization error is negligible. This is exactly how we convert the state  $x_i$  of a node from real value to a fixed length integer and back to a floating point number. The conversion is performed at each iteration of the protocol.

### B. Subtraction and Negative Values

Another issue is how to treat the sign of an integer for encryption. [19] solves this problem by mapping negative values to the end of the group  $\mathbb{Z}_n$  where  $n = pq$  is given by the public key. We offer an alternative solution by taking advantage of the fact that encryption algorithms blindly treat bit strings as an unsigned integer. In our implementation all integer values are stored in fix-length integers (i.e., `long int` in C) and negative values are left in two's complement format. Encryption and intermediate computations are carried out as if the underlying data were unsigned. When the final message is decrypted, the overflowed bits (bits outside the fixed length) are discarded and the remaining binary number is treated as a signed integer which is later converted back to a real value.

## VIII. NUMERICAL SIMULATIONS AND HARDWARE EXPERIMENTS

In this section, we first numerically verify effectiveness of the proposed approach and show its advantage over existing state-of-the-art approaches. Then we provide hardware experimental results on a resource-constrained Raspberry-Pi board based micro-controller network to demonstrate the efficiency of the proposed approach.

### A. Numerical Implementation of the Proposed Approach

To illustrate the capability of our protocol, we implemented the consensus protocol in C/C++. We used an open-source C implementation of the Paillier cryptosystem [41] because it allowed byte-level access. For each exchange between two nodes, the states were converted to 64-bit integers by multiplying  $Q = 10^5$ . The weights  $a_{i \rightarrow j}^{(k)}$  were also scaled up similarly and represented by 64-bit random integers. The encryption/decryption keys were set to 256-bit long.

The first simulation had four nodes connected in a undirected ring, which led to  $\Delta = \max_i |N_i| = 2$ . We set the step-size to  $\varepsilon = 0.5$ . Given  $0 < a < \bar{a} < \frac{1}{\sqrt{\varepsilon\Delta}} = \frac{1}{\sqrt{0.5*2}} = 1$ , we set the admissible range to  $[a, \bar{a}] = [0.01, 0.99]$ . The initial states were set to  $\{1, 2, 4, 8\}$  respectively and the average is 3.75. Each node used a static key pair which was initialized once at the beginning. The states' convergence to the average is shown in Fig. 4.

The plot of received encrypted messages which encoded the weighted difference between two nodes is given in Fig. 5. It is worth noting that although the states have converged to the average, the encrypted weighted differences still appear to be random to an unintended observer.

The second simulation considered the weighted average consensus. Under the same initial condition as the first example, the nodes also had the associated weights  $\{0.1, 0.2, 0.3, 0.4\}$ . The step size  $\varepsilon$  was set to 0.05. Under

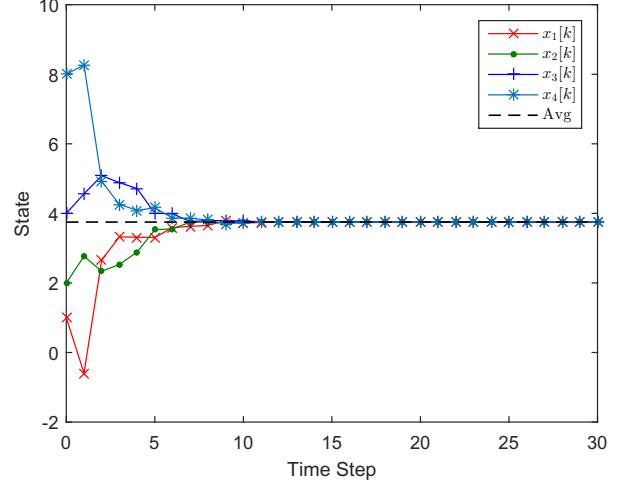


Fig. 4. Convergence to the average consensus. The states converge to the average consensus value 3.75 in about 10 steps.

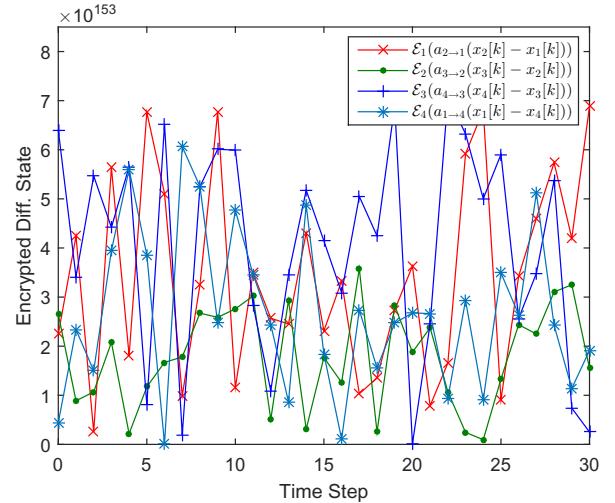


Fig. 5. Encrypted weighted differences vs time step. Although the states have converged after 10 steps, the encrypted differences still appear to be random.

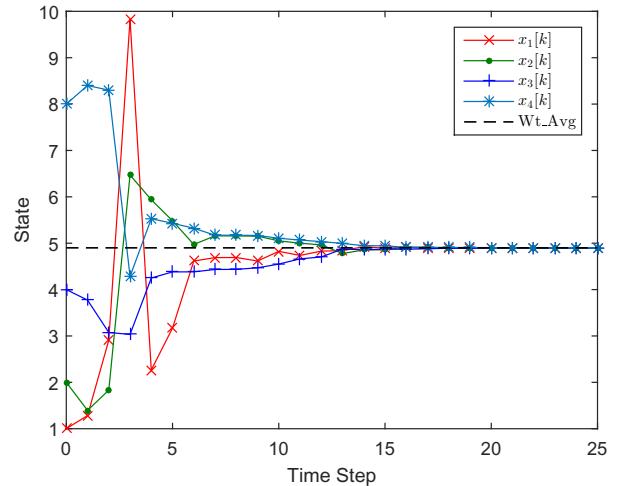


Fig. 6. Convergence to the weighted average consensus. The states converge to the weighted average consensus value 4.9 in about 15 steps.

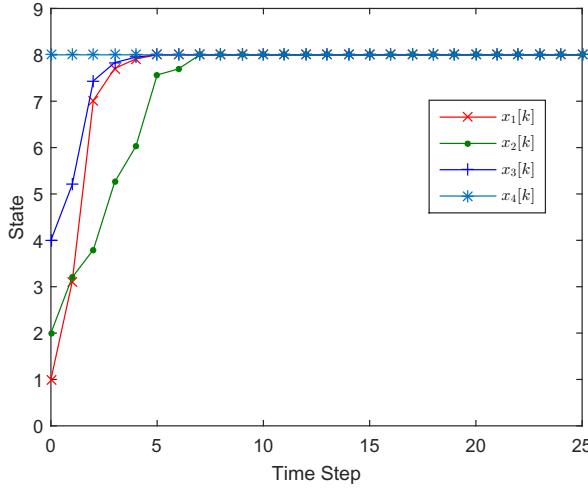


Fig. 7. Convergence to the maximum consensus. The states converge to the maximum consensus value 8 in less than 10 steps.

$0 < \underline{a} < \bar{a} < \sqrt{\frac{\min_i w_i}{\varepsilon \Delta}} = \sqrt{\frac{0.1}{0.05*2}} = 1$ , we set the admissible range to  $[\underline{a}, \bar{a}] = [0.01, 0.99]$ . Fig. 6 shows that the states converge to the weighted average value 4.9.

The third simulation considered the maximum consensus, using the same graph structure and initial condition. Fig. 7 shows that the states converge to the exact maximum in about 7 steps.

The computational overhead caused by the encryption is manageable. Without any hardware-specific optimization, it takes about 7 ms to compute one exchange of state on a desktop computer with a 3.4 GHz CPU and 4.00 GB memory.

### B. Comparison with Existing Results

In this subsection, we show the advantages of the proposed average consensus approach over existing data-obfuscation based average consensus approaches, more specifically, the decaying-noise approach by Mo and Murray [12], the finite-noise-sequence approach by Manitara and Hadjicostis [13], and the differential-privacy based approach by Huang *et al.* [11]. The differential-privacy based approach in [11] injects uncorrelated Laplace noise to exchanged states to ensure the privacy of participating nodes and is subject to a trade-off between privacy and accuracy. To guarantee the accuracy of average consensus, the approaches in [12] and [13] employ correlated noises, which, however, compromise the achievable privacy. In this subsection, we use numerical simulations to show that all these approaches are vulnerable to an external eavesdropper.

In the numerical simulation, we used the network topology and weight matrix provided in [12], i.e.,

$$\mathbf{A} = \frac{1}{4} \begin{bmatrix} 2 & 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 3 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (47)$$

We set the initial states of five nodes to  $\{1, 2, 3, 4, 5\}$  respectively.

We consider the case where node 1's communication with all its neighbors is tapped by an external eavesdropper Eve. Eve also knows the internal dynamics and network interaction topology, i.e., the  $\mathbf{A}$  matrix.

1) *Comparison with the approach in [12]:* We first simulated the approach in [12]. For consistency and ease of reading, we borrow the notations from [12], where the authors denote the internal state and the transmitted state by  $x_i[k]$  and  $x_i^+[k]$ , respectively, i.e.,

$$x_i^+[k] = x_i[k] + w_i[k] \quad (48)$$

where  $w_i[k]$  is a random noise specified as follows

$$w_i[k] = \begin{cases} v_i[0], & \text{if } k = 0 \\ \rho^k v_i[k] - \rho^{k-1} v_i[k-1] & \text{otherwise} \end{cases} \quad (49)$$

In (49),  $\rho \in (0, 1)$  holds and  $v_i[k]$  is a normally distributed random variable with zero mean and unit variance.  $\{v_i[k]\}_{i=1, \dots, N, k=0, 1, \dots}$  are jointly independent but from (49), it is clear that the injected noise  $w_i[k]$  is time-correlated.

Each node updates its internal state by

$$x_i[k+1] = a_{ii}x_i^+[k] + \sum_{v_j \in N_i} a_{ij}x_j^+[k] \quad (50)$$

Based on the assumption that Eve has access to the in-bound and out-bound messages of node 1 (i.e.,  $x_1^+[k]$  and  $x_j^+[k]$  for  $v_j \in N_1$ ) as well as the interaction dynamics/topology (i.e.,  $a_{1j}$ ), we can build an observer to estimate node 1's initial state as follows:

- The initial state of the observer is set to the first out-going transmission from node 1, i.e.

$$z[0] = x_1^+[0] \quad (51)$$

- The update of the observer is based on accumulating the difference between the transmitted state and the predicted state

$$z[k+1] = z[k] + x_1^+[k+1] - (a_{11}x_1^+[k] + \sum_{v_j \in N_1} a_{1j}x_j^+[k]) \quad (52)$$

It is shown in Fig. 8 that as the network converges and all node states approach the average consensus value, the observer also converges to node 1's true initial value.

In contrast, with our method all the messages are encrypted by design. Consequently, Eve cannot gain any advantage by wiretapping the communication.

2) *Comparison with the approach in [13]:* Although using a different noise sequence, the approach in [13] builds on a similar noise-injection approach to that of [12] (in the sense that later noise will cancel out previous noise). Therefore, we used the same observer design as in the simulation of [12]. Again we obtained that as the network converged, the observer value converged to node 1's true initial state, as depicted in Fig. 9.

On the contrary, such external eavesdropper gains no information from our protocol where transmitted messages are encrypted inherently.

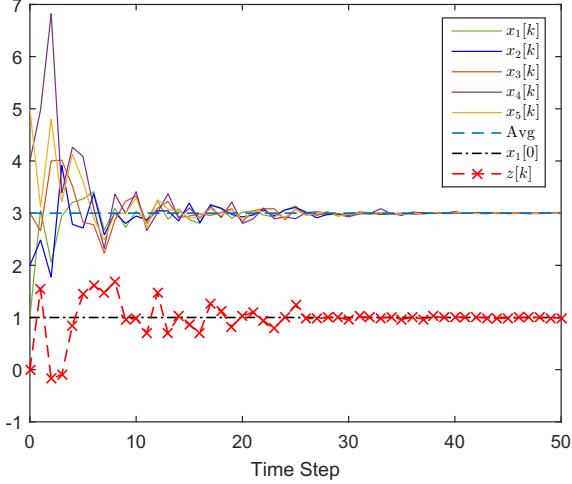


Fig. 8. If Eve wiretaps all the messages of node 1 and has access to node dynamics and interaction topology, then the initial state of node 1 can be estimated under the privacy protocol in [12].

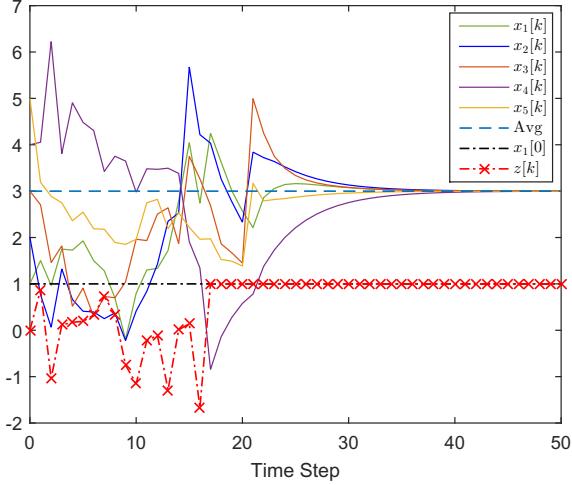


Fig. 9. If Eve wiretaps all the messages of node 1 and has access to node dynamics and interaction topology, then the initial state of node 1 can be estimated under the privacy protocol in [13].

3) *Comparison with the approach in [11]:* We next show the vulnerability of the approach in [11] to an external eavesdropper. We built a naive estimator that guesses the initial state by taking the value of the initial out-bound message of the node. Using the notations from [11], the estimated state is given by

$$z = x_1[0] = \theta_1[0] + \eta_1[0] \quad (53)$$

where  $x_1$  is the sent message,  $\theta_1[0]$  is the true initial state, and  $\eta_1[0]$  is a random noise generated from a Laplace distribution  $\text{Lap}(cq^0)$ .

We then varied the noise parameter  $c$  and repeated the simulation for 500 times. The result is summarized in Fig. 10 with  $\text{Avg\_Err}$  and  $\text{Est\_Err}$  denoting error in consensus value achieving and initial state estimation respectively. We can conclude that although the differential-privacy based approach can use a large noise level to prevent an external eavesdropper

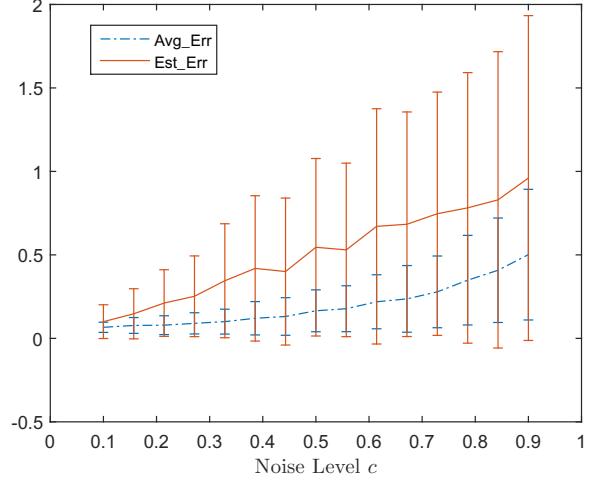


Fig. 10. Although the differential-privacy based approach guarantees that the initial state cannot be estimated exactly, the accuracy of the final consensus value and the accuracy of the estimated initial value are correlated.

from accurately inferring the initial state, the noise will also lead to a large error in the final consensus value. As a result, when an application calls for higher accuracy of the consensus result, the risk of disclosing one's initial state also becomes higher.

Our method protects the privacy by an encrypted message-exchange mechanism that does not affect the accuracy of the final consensus value. Therefore the user does not have to worry about the trade-off between accuracy and privacy.

### C. Hardware Experiments

To confirm the efficiency of the secure and privacy-preserving average consensus approach in real-world cyber-physical systems, we also implemented the proposed approach on six Raspberry Pi boards with 64-bit ARMv8 CPU and 1 GB RAM (cf. Fig. 11, source code available at [42]).

In the implementation, the communication was conducted through Wi-Fi based on the “sys/socket.h” C library. Paillier encryption/decryption was realized using the “libpaillier-0.8” library from [43]. To obtain  $\Delta x_{ij}$  in a pairwise interaction, a node employs a request message to initialize the interaction and the other node replies with a response message. In a multi-node network, for a node to be able to simultaneously receive requests and responses from multiple neighbors, parallelism needed to be introduced. The “pthread” C library was used to generate multiple parallel threads to handle incoming requests and responses. Each time a node receives a request/response, it generates a new thread to handle it and immediately listens for more requests. Because in the implementation, it is impossible to start all nodes simultaneously, a counter is introduced on each node and its value is embedded in each request/response packet to help nodes make sure that they are on the same pace. For 64 byte encryption key, the size of the actual packet is 144 bytes, which includes all necessary headers and stuffing bytes. For each interaction, the average processing latency was 7.8 ms, which is acceptable for most real-time control systems.

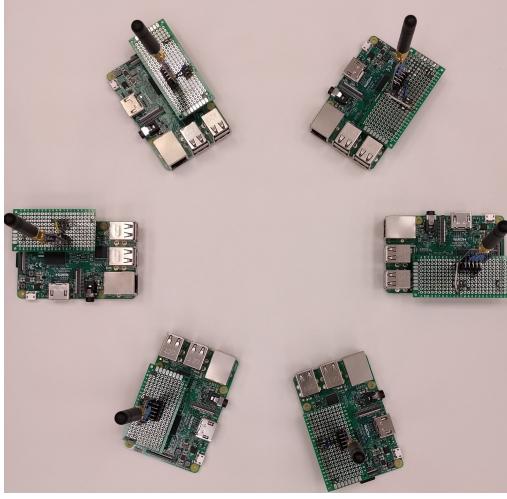


Fig. 11. A network of six Raspberry Pi boards.

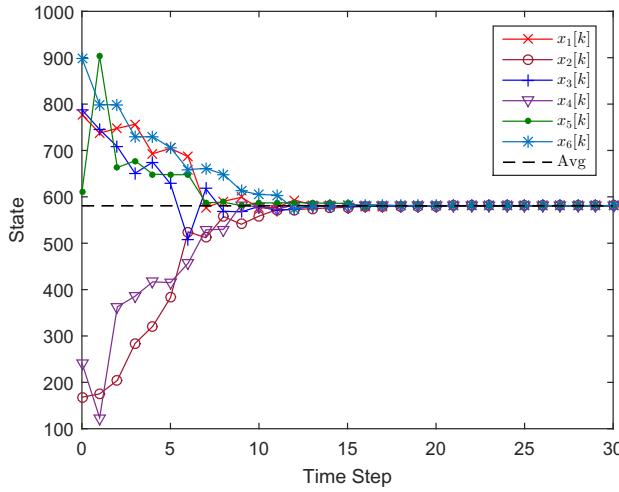


Fig. 12. All nodes converge to the average consensus value in the experimental verification using Raspberry Pi boards. The states have initial values as 777, 168, 788, 242, 610, and 899, respectively and they all converge to the average consensus value 580.67 in about 13 steps.

The implementation result is given in Fig. 12, which shows that perfect consensus can be achieved.

## IX. DISCUSSIONS AND CONCLUSIONS

In this paper we proposed a decentralized secure and privacy-preserving protocol for the network consensus problem which can guarantee the security and privacy of a node as long as it has at least one legitimate neighbor. In contrast to existing data-obfuscation based average consensus approaches which are vulnerable to an external eavesdropper, we encode randomness into the system dynamics with the help of an additive homomorphic cryptosystem which allows a deterministic convergence to the average (subject to a quantization error). The protocol also provides resilience to passive attackers and allows easy incorporation of active attacker defending mechanisms. In addition to average consensus, our protocol can

be easily extended to enable security and privacy in weighted average consensus and maximum/minimum consensus under new update rules. Both numerical simulations and hardware experiments are provided to demonstrate the effectiveness and efficiency of the approach.

The proposed approach offers several advantages over existing privacy-preserving average consensus protocols using noise-injection based data obfuscation, such as the uncorrelated-noise (differential-privacy) based approach in [11] and the correlated-noise based approaches in [12], [13]. First, compared with the differential-privacy based approach, our approach can guarantee convergence to the consensus value (subject to a quantization error) in a deterministic manner. Secondly, our approach encrypts the exchanged messages and is resilient to external eavesdroppers wiretapping the communication channel, whereas an external eavesdropper can easily defeat the approaches in [12] and [13] by using a naive observer, as confirmed by the numerical simulation results in Sec. VIII-B. Furthermore, the proposed approach is naturally extendable to time-varying networks, whereas existing noise-injection based data obfuscation approaches all assume time-invariant parameters, which could be troublesome if the topology of the network or the number of nodes is not constant.

On the other hand, although the computational overhead introduced by the encryption algorithm (7 ms in numerical simulations and 7.8 ms in hardware experiments) is acceptable in most real-time control systems, it is indeed higher compared to the unencrypted alternatives. We argue that the benefits of using encryption to preserve privacy and security outweigh the computational burden which is easily manageable on modern micro-controllers.

## ACKNOWLEDGEMENT

The authors would like to thank Christoforos Hadjicostis and Yilin Mo for their comments on an initial draft of this article.

## REFERENCES

- [1] M. Ruan, M. Ahmaz, and Y. Q. Wang. Secure and privacy-preserving average consensus. In *Proceedings of the 3rd ACM workshop on Cyber-Physical Systems Security and Privacy*, 2017.
- [2] M. H. Degroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
- [3] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.
- [4] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [5] J. N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, 1984.
- [6] Z. Zhang and M. Y. Chow. Incremental cost consensus algorithm in a smart grid environment. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–6, July 2011.
- [7] E. Nozari, P. Tallapragada, and J. Cortés. Differentially private average consensus with optimal noise selection. *IFAC-PapersOnLine*, 48(22):203 – 208, 2015.
- [8] Z. Huang, S. Mitra, and N. Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, ICDCN ’15, New York, NY, USA, 2015. ACM.
- [9] E. Nozari, P. Tallapragada, and J. Cortés. Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design. *Automatica*, 81(7):221–231, 2017.

[10] V. Katewa, F. Pasqualetti, and V. Gupta. On privacy vs cooperation in multi-agent systems. *International Journal of Control*, (accepted):1–20, 2017.

[11] 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*, pages 81–90. ACM, 2012.

[12] Y. Mo and R. M. Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765, Feb 2017.

[13] N. E. Manitara and C. N. Hadjicostis. Privacy-preserving asymptotic average consensus. In *2013 European Control Conference (ECC)*, pages 760–765, July 2013.

[14] J. He, L. Cai, P. Cheng, M. Xing, J. Pan, and L. Shi. Private and accurate data aggregation against dishonest nodes. *arXiv preprint arXiv:1609.06381*, 2016.

[15] S. S Kia, J. Cortés, and S. Martínez. Dynamic average consensus under limited control authority and privacy requirements. *International Journal of Robust and Nonlinear Control*, 25(13):1941–1966, 2015.

[16] S. Pequito, S. Kar, S. Sundaram, and A. P. Aguiar. Design of communication networks for distributed computation with privacy guarantees. In *In Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 1370–1376, 2014.

[17] A. Alaeddini, K. Morgansen, and M. Mesbah. Adaptive communication networks with privacy guarantees. In *Proceedings of 2017 American Control Conference*, pages 4460–4465, 2017.

[18] N. Gupta and N. Chopra. Confidentiality in distributed average information consensus. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6709–6714, Dec 2016.

[19] K. Kogiso and T. Fujita. Cyber-security enhancement of networked control systems using homomorphic encryption. In *Proceedings of the 54th IEEE International Conference on Decision and Control*, pages 6836–6843, 2015.

[20] Y. Shoukry, G. Konstantinos, 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 55th IEEE International Conference on Decision and Control*, pages 5053–5058, 2016.

[21] 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 Processing Magazine*, 30(1):82–105, Jan 2013.

[22] 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)*, pages 7406–7410, May 2014.

[23] N. M. Freris and P. Patrinos. Distributed computing over encrypted data. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1116–1122, Sept 2016.

[24] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[25] T. ElGamal. *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, pages 10–18. Springer Berlin Heidelberg, Berlin, Heidelberg, 1985.

[26] P. Paillier. *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, pages 223–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[27] J. Katz and Y. Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman & Hall/CRC, Boca Raton, 2008.

[28] O. Goldreich. *Foundations of Cryptography: volume 2, Basic Applications*. Cambridge University Press, 2001.

[29] O. Goldreich. *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2007.

[30] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, Feb 2005.

[31] A. Nedić, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[32] A. Nedić, A. Olshevsky, and W. Shi. A geometrically convergent method for distributed optimization over time-varying graphs. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1023–1029. IEEE, 2016.

[33] A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[34] K. Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Trans. Knowl. Data Eng.*, 18(1):92–106, 2006.

[35] S. Han, W. K. Ng, L. Wan, and V. C. S. Lee. Privacy-preserving gradient-descent methods. *IEEE Trans. Knowl. Data Eng.*, 22(6):884–899, 2010.

[36] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Trans. Parallel Distrib. Syst.*, 25(1):222–233, 2014.

[37] A. G. Konheim. *Computer security and cryptography*. John Wiley & Sons, 2007.

[38] O. Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[39] W. Zhang, Z. Wang, Y. Guo, H. Liu, Y. Chen, and J. Mitola III. Distributed cooperative spectrum sensing based on weighted average consensus. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.

[40] X. Duan, J. He, P. Cheng, Y. Mo, and J. Chen. Privacy preserving maximum consensus. In *In Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 4517–4522, 2015.

[41] J. Bethencourt. Advanced crypto software collection. <http://acsc.cs.utexas.edu/libpaillier/>.

[42] *Source Code for Confidential Average Consensus*. <https://github.com/Yongqiang-Wang/Raspberry-Paillier/>.

[43] *Paillier Library*. <http://acsc.cs.utexas.edu/libpaillier/>.



**Minghao Ruan** received the B.S. degree in Electrical Engineering from Syracuse University in Syracuse, New York, USA, in 2012; and M.S. degree in Robotics from the Robotic Institute of Carnegie Mellon University in Pittsburgh, Pennsylvania, USA, in 2014. He is currently a PhD student of the Department of Electrical Engineering of Clemson University.



**Huan Gao** was born in Shandong, China. He received the B.S. degree in automation and M.Sc. degree in control theory and control engineering from Northwestern Polytechnical University, Shaanxi, China, in 2011 and 2015, respectively. Currently, he is working toward the Ph.D. degree at Clemson University. His research focuses on cooperative control and dynamics of pulse-coupled oscillators.



**Yongqiang Wang** was born in Shandong, China. He received the B.S. degree in electrical engineering and automation, the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2004, and the M.Sc. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2009. From 2007 to 2008, he was with the University of Duisburg-Essen, Germany, as a Visiting Student. He was a Project Scientist with the University of California at Santa Barbara, Santa Barbara. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Clemson University. His research interests are cooperative and networked control, synchronization of wireless sensor networks, systems modeling and analysis of biochemical oscillator networks, and model-based fault diagnosis.

Dr. Wang was a recipient of the 2008 Young Author Prize from the IFAC Japan Foundation for a paper presented at the 17th IFAC World Congress in Seoul.