Secure and Privacy-Preserving Consensus

Minghao Ruan, Huan Gao, and Yongqiang Wang

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 that enables 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 consensus to the exact value in a deterministic manner without disclosing a node’s state to its neighbors. 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 which 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-preservation approach is also applicable to weighted average consensus as well as maximum/minimum consensus under a new update rule. The approach is light-weight in computation and communication. Implementation details and numerical examples are provided to demonstrate the capability of our approach.

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 [1], [2]. 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 [3].

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 secret their individual personal opinions [4]. Another example is power systems where multiple generators want to reach agreement on cost while keeping their individual generation information private [5]. Secondly, storing or exchanging information in the unencrypted plaintext form is vulnerable to attackers which try to 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 [6], [7], [8], [9], [10], [11], [12], [13]. Most of these approaches use the idea of obfuscation to mask the true state values by adding carefully-designed noise on the state. Such approaches usually exploit tools such as mean-square statistics or differential privacy which is heavily used for database privacy in computer science [7], [8], [9], [10], [11], [12]. Although they enhance privacy, such noise-based obfuscation also unavoidably affects the performance of consensus, either directly preventing converging to the true value, or making convergence only achievable in the statistical mean-square sense. 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 [14], [15], [16]. 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 which 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 [17], like in cloud-based control or computation [18], [19], [20], 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 fact, to our knowledge, in the only two reported efforts incorporating cryptography into decentralized average consensus [21], [22], privacy is obtained by paying the price of depriving participating nodes from access to the final consensus value (note that in [22] individual participating nodes do not have the decryption key to decrypt the final consensus value which is in the encrypted form, otherwise they
and connected. Therefore, throughout this paper we assume that the graph is undirected.

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 is the greatest common divisor of $m$ and $n$, where $gcd(a, b)$ is the greatest common divisor of $a$ and $b$. The plaintext is $m = L(c^2 \mod n^2) \cdot \mu \mod n$.

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., $E(m_1 + m_2)$, can be obtained based on $E(m_1)$ and $E(m_2)$ directly:

$$E(m_1, r_1) \cdot E(m_2, r_2) = (g^{m_1 r_1} \cdot g^{m_2 r_2} \mod n^2) \mod n^2$$

$$= (g^{m_1 + m_2} \cdot (r_1 r_2)^n) \mod n^2$$

$$= E(m_1 + m_2, r_1 r_2)$$
The dependency on random numbers $r_1$ and $r_2$ is explicitly shown in (3), yet they play no role in the decryption. Therefore, the following shorthand notation will be used instead:

$$E(m_1) \cdot E(m_2) = E(m_1 + m_2) \quad (6)$$

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

$$E(m)^k = \sum_{i=1}^{k} E(m) = E(\sum_{i=1}^{k} m) = E(km) \quad (7)$$

Notice however, the Paillier cryptosystem is not multiplicative homomorphic because $k$ in (7) is in the plaintext form. Furthermore, the existence of the random number $r$ in Paillier cryptosystem gives it resistance to dictionary attacks [26] 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 $N$ 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 $a_{ij}^{(t)}$, 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 exact average value.

In this section we present details of our confidential interaction protocol based on (3) and (4). In particular we show how a node can obtain the weighted difference (8) between itself and any of its neighbors without disclosing each other’s state information:

$$\Delta x_{ij} = a_{ij}^{(t)} \cdot (x_j - x_i)$$

$$\Delta x_{ji} = a_{ji}^{(t)} \cdot (x_i - x_j) \quad (8)$$

subject to $a_{ij}^{(t)} = a_{ji}^{(t)} > 0$

Plugging the state difference (3) into (8) gives a new formulation of continuous-time average consensus

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} \Delta x_{ij}(t) \quad (9)$$

Similarly, we can rewrite the discrete-time consensus update rule as

$$x_i[k+1] = x_i[k] + \varepsilon \sum_{j \in \mathcal{N}_i} \Delta x_{ij}[k] \quad (10)$$

Notice that in a decentralized system it is impossible to protect the privacy of both nodes in a pairwise interaction if the protocol (3) is used without a third party distributing secret $a_{ij}^{(t)}$. 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 $a_{ij}^{(t)}$, it can still infer the value of $x_j(t)$ through $x_j = \frac{\Delta x_{ij}^*}{a_{ij}^{(t)}} + x_i$.

From now on, for the sake of simplicity in bookkeeping, we omit the superscript $t$ in $a_{ij}^{(t)}$. 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 $a_{ij} = a_i \cdot a_j = a_{ji}$, with $0 \leq a_i, a_j \leq \tilde{a}$ only known to node $v_i$ and $a_j \geq 0$ only known to node $v_j$ (here $\tilde{a}$ is a positive value denoting the range in implementations which will be explained in detail later). 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. 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 $(k_{pq}, k_{qs})$, $i \in \{1, 2\}$.

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 non-negative random number $a_1$ (resp. $a_2$) which is within a certain range $[0, \tilde{a}]$ in implementation. First, node $v_1$ sends its encrypted negative state $E_1(-x_1)$ as well as the public key $k_{p1}$ to node $v_2$. Note that here the subscript in $E_1$ denotes encryption using the public key of node $v_1$. Node $v_2$ then computes the encrypted $a_2$-weighted difference $E_1(a_2(-x_2))$ following the three steps below:

1) Encrypt $x_2$ with $v_1$’s public key $k_{p1}$: $x_2 \rightarrow E_1(x_2)$.

2) Compute the difference directly in ciphertext:

$$E_1(x_2 - x_1) = E_1(x_2 + (-x_1)) = E_1(x_2) \cdot E_1(-x_1) \quad (11)$$

3) Compute the $a_2$-weighted difference in ciphertext:

$$E_1(a_2 \cdot (x_2 - x_1)) = (E_1(x_2 - x_1))^{a_2} \quad (12)$$

Then $v_2$ returns $E_1(a_2 \cdot (x_2 - x_1))$ to $v_1$. After receiving $E_1(a_2 \cdot (x_2 - x_1))$, $v_1$ decrypts it using the private key $k_{s1}$ and multiplies the result with $a_1$ to get the weighted difference $\Delta x_{12}$:

$$E_1(a_2(x_2 - x_1)) \xrightarrow{D_1} a_2(x_2 - x_1) \quad \Delta x_{12} = a_1a_2(x_2 - x_1) \quad (13)$$

In a similar manner, the exchange $v_2 \rightarrow v_1 \rightarrow v_2$ produces $E_2(a_1(x_1 - x_2))$ for $v_2$ who then decrypts the message and multiplies the result by its own multiplier $a_2$ to get $\Delta x_{21}$:

$$E_2(a_1(x_1 - x_2)) \xrightarrow{D_2} a_1(x_1 - x_2) \quad \Delta x_{21} = a_2a_1(x_1 - x_2) \quad (14)$$

After each node collects the weighted differences from all neighbors, it updates its state with (3) or (4) accordingly.

Several remarks are in order:

- The construction of each $a_{ij}$ as the product of two random numbers $a_i$ and $a_j$ is key to guarantee that the weights are symmetric, i.e., $a_{ij} = a_{ji}$, which is important for average consensus.

- $v_2$ does not have the private key of $v_1$ and cannot see $x_1$ which is encrypted in $E_1(-x_1)$.

- Given $a_2(x_2 - x_1)$, $v_1$ cannot solve for $x_2$ because $a_2$ is only known to $v_2$. 
The network dynamics in (3) can be rewritten as:

\[ a \text{to set the appropriate values for the multiplier } a \text{ in ciphertext. Note that } \]

\[ \text{via a communication channel. Shaded nodes indicate the computation done} \]

\[ \text{Fig. 1. A step-by-step illustration of the confidential interaction protocol.} \]

\[ \text{Initial State} \]

\[ \begin{align*}
&v_1, a_1, (k_{p1}, k_{s1}), \\
&v_2, a_2, (k_{p2}, k_{s2}).
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&\mathcal{E}(x_{1}, k_{p1}) \\
&\mathcal{E}(x_{2}, k_{p2}).
\end{align*} \]

\[ \text{Encrypt the Negative} \]

\[ \text{State (with its own key)} \]

\[ \begin{align*}
&\mathcal{E}_1(-x_1) \\
&\mathcal{E}_2(-x_2).
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&\mathcal{E}(x_{1}, k_{p1}) \\
&\mathcal{E}(x_{2}, k_{p2}).
\end{align*} \]

\[ \text{Transmit the State} \]

\[ \text{and Public Key} \]

\[ \begin{align*}
&\mathcal{E}_2(-x_2), k_{p2} \\
&\mathcal{E}_1(-x_1), k_{p1}.
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&\mathcal{E}(x_{1}, k_{p1}) \\
&\mathcal{E}(x_{2}, k_{p2}).
\end{align*} \]

\[ \text{Encrypt the State} \]

\[ \text{(with received key)} \]

\[ \begin{align*}
&\mathcal{E}_3(x_1) \\
&\mathcal{E}_4(x_2).
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&\mathcal{E}_2(x_1 - x_2) \\
&\mathcal{E}_1(x_2 - x_1).
\end{align*} \]

\[ \text{Compute the Difference} \]

\[ \text{(in ciphertext)} \]

\[ \begin{align*}
&\mathcal{E}_2(x_1 - x_2) \\
&\mathcal{E}_1(x_2 - x_1).
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&\mathcal{E}_2(x_1 - x_2)^{\ast} \\
&\mathcal{E}_1(x_2 - x_1)^{\ast}.
\end{align*} \]

\[ \text{Multiply the Weight} \]

\[ \text{(in ciphertext)} \]

\[ \begin{align*}
&\mathcal{E}_2(a_1(x_1 - x_2)) \\
&\mathcal{E}_1(a_2(x_2 - x_1)).
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&D_1(-, k_{s1}) \\
&D_2(-, k_{s2}).
\end{align*} \]

\[ \text{Transmit the Result} \]

\[ \text{Back to Sender} \]

\[ \begin{align*}
&a_2(x_2 - x_1) \\
&a_1(x_1 - x_2).
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&\mathcal{D}_1(-, k_{s1}) \\
&\mathcal{D}_2(-, k_{s2}).
\end{align*} \]

\[ \text{Decrypt the Result} \]

\[ \begin{align*}
&a_2(x_2 - x_1) \\
&a_1(x_1 - x_2).
\end{align*} \]

\[ \rightarrow \]

\[ \begin{align*}
&\Delta x_{12} = a_1 a_2 (x_2 - x_1) \\
&\Delta x_{21} = a_2 a_1 (x_2 - x_1).
\end{align*} \]

\[ \text{Multiply the Weight} \]

\[ \text{(in plaintext)} \]

\[ \begin{align*}
&\Delta x_{12} = a_1 a_2 (x_2 - x_1) \\
&\Delta x_{21} = a_2 a_1 (x_2 - x_1).
\end{align*} \]

\[ \text{where } \mathbf{L}^{(t)} = [l^{(t)}_{ij}] \text{ is the time-varying Laplacian matrix defined by} \]

\[ l^{(t)}_{ij} = \begin{cases} 
\sum_{j \in N_i} a^{(t)}_{ij} & i = j \\
-a^{(t)}_{ij} & i \neq j 
\end{cases} \] \hfill (16)

\[ \text{Theorem 1:} \text{ If the couplings weights } a^{(t)}_{ij} \text{ in (16) are established according to the confidential interaction protocol in Sec. III, then under any positive bound } \bar{a} > 0, \text{ the system will achieve average consensus with states converging to} \]

\[ \lim_{t \to \infty} x(t) = \mathbf{1} \alpha \text{ where } \alpha = \text{Avg} (0) = \frac{1}{n} \mathbf{1}^T x(0) \] \hfill (17)

\[ \text{Proof:} \text{ It is already known that average consensus can be achieved if for all time } t_0 > 0, \text{ there exists a constant } T > 0 \text{ such that } a^{(t)}_{ij} > 0 \text{ is true for some } t \in [t_0, t_0 + T] \text{ [3], [27]. Noting that the weights } a^{(t)}_{ij} \geq 0 \text{ obtained from the confidential interaction protocol in Sec. III are random and independent of each other, the proof can be obtained by following the line of reasoning in [3], [27].} \]

\[ \text{B. Convergence for Discrete Time Consensus} \]

\[ \text{In discrete-time domain} \text{ can be rewritten as} \]

\[ x[k+1] = \mathbf{P}^{(k)} x[k] \] \hfill (18)

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

\[ l^{(k)}_{ij} = \begin{cases} 
\sum_{j \in N_i} a^{(k)}_{ij} & i = j \\
-a^{(k)}_{ij} & i \neq j 
\end{cases} \] \hfill (19)

\[ \text{Theorem 2:} \text{ If the couplings weights } a^{(k)}_{ij} \text{ in (19) are established according to the confidential interaction protocol in Sec. III and } \varepsilon \text{ satisfies } 0 < \varepsilon < \frac{1}{2} \text{ where } \Delta = \max |N_i| \text{ with } | \cdot | \text{ denoting the set cardinality, then under any positive bound } 0 < \bar{a} < 1, \text{ the system will achieve average consensus with states converging to} \]

\[ \lim_{k \to \infty} x[k] = \mathbf{1} \alpha \text{ with } \alpha = \text{Avg} (0) = \frac{1}{n} \mathbf{1}^T x(0) \] \hfill (20)

\[ \text{Proof:} \text{ The proof can be obtained by following the similar line of reasoning of Corollary 2 in [3].} \]

\[ \text{Remark 1:} \text{ Since the framework allows time-varying weighted matrix } \mathbf{A}^{(t)} \text{ for the continuous time domain or } \mathbf{A}^{(k)} \text{ for the discrete time domain, it can easily be extended to the case with switching interaction graphs according to [27].} \]

\[ \text{V. Analysis of Privacy and Security} \]

\[ \text{Privacy and security are often used interchangeably in the literature but here we make the distinction explicit. Among the control community privacy is equivalent to the concept of unobservability. Privacy is also closely related to the concept of semantic security from cryptography [25]. Both concepts essentially concern with an honest-but-curious adversary which is interested in learning the states of the network but conforms to the rules of the system. Security, on the other hand, deals with a broader issue which includes learning the states as well as the possibilities of exploiting the system to cause damages.} \]
collected by Eve are given by $A$ (Alice) and $\overline{\gamma}$ (other nodes) to name the legitimate sender and receiver participants as explained in Theorem 4). The steps (except in a trivial case that should always be avoided, an honest-but-curious node Eve cannot learn 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 in which it is possible for Eve to infer other nodes’ states which should be avoided.

**Remark 2:** Following the same line of reasoning, it can be seen that Eve cannot establish privacy leak.

**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 the communication.

The Paillier encryption algorithm is known to provide semantic security, i.e., Indistinguishability under Chosen Plaintext Attack (IND-CPA) [25]. As a result, the recipient of the first transmission $E(-x_i)$ cannot see the value of $x_i$ at all time. We now prove that an honest-but-curious adversary cannot infer the initial state of a neighbor even if it can accumulate and correlate the return messages $a_i a_j (x_j - x_i)$ in multiple steps (except in a trivial case that should always be avoided, as explained in Theorem 4).

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

**Theorem 3:** Assume all nodes follow the confidential interaction protocol. An honest-but-curious node Eve cannot learn the initial state of a neighboring node 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 information accessible to Eve. From the perspective of the honest-but-curious node Eve, the measurements seen at each time step $k$ are $\Delta x_{EA} = a_i^{(k)} a_E^{(k)} (x_i[k] - x_E[k])$, $i \in \{A, B\}$. In matrix form, define these information as $y_E[k]$

\[
y_E[k] = \Delta x_{EA}, \Delta x_{EB} = C_E^{(k)} x[k]
\]

where:

\[
C_E^{(k)} = \begin{bmatrix}
-a_A^{(k)} a_E^{(k)} & a_A^{(k)} a_E^{(k)} & 0 \\
-a_B^{(k)} a_E^{(k)} & a_B^{(k)} a_E^{(k)} & 0
\end{bmatrix}
\]

It can be easily derived that after $K$ steps, the measurements collected by Eve are given by

\[
y_E[0 : K] = O_{E,[0:K]} x[0]
\]

where the observability matrix $O_{E,[0:K]}$ is given by

\[
O_{E,[0:K]} = \begin{bmatrix}
C_E^{(0)} \\
C_E^{(1)} P^{(0)} \\
\vdots \\
C_E^{(K)} \prod_{k=K-1}^{0} P^{(k)}
\end{bmatrix}
\]

with $P^{(k)}$ being the Perron matrix defined in [18]. Note that the entries of $C_E^{(k)}$ and $P^{(k)}$ are unknown to Eve because $a_A^{(k)}$ and $a_B^{(k)}$ are randomly chosen by Alice and Bob respectively. Therefore, the ability for Eve to infer the state of other nodes cannot be analyzed using conventional observability based approach in e.g., [15, 16].

We propose a new analysis approach based on the solvability of systems of equations. From [22] it can be seen that Eve can establish $2(K+1)$ equations based on received information from time instant 0 to $K$. Given that after consensus, Eve can know the final state of other nodes which is equal to its own final state (represent it as $x_{\text{consensus}}$), it can establish one more equation

\[
x_A[0] + x_B[0] + x_E[0] = 3x_{\text{consensus}},
\]

which makes the number of employable equations to $2(K+1) + 1$. If there are more than $2(K+1) + 1$ unknowns involved in these $2(K+1) + 1$ equations, then it is safe to say that Eve cannot solve the equations and get the initial states of $x_A[0]$ and $x_B[0]$. In fact, the confidential interaction protocol introduces $2(K+1)$ unknown parameters $a_A^{(0)}, a_A^{(1)}, \cdots, a_A^{(K)}, a_B^{(0)}, a_B^{(1)}, \cdots, a_B^{(K)}$, which, in combination with $x_A[0], x_B[0]$ unknown to Eve, will make the total number of unknowns to $2(K+1)+2$. Therefore, the honest-but-curious Eve cannot use the accessible $2(K+1)+1$ system of equations in [22] to solve for the initial states of $x_A[0]$ and $x_B[0]$. 

**Remark 2:** 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 in which it is possible for Eve to infer other nodes’ states which should be avoided.

**Theorem 4:** 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 inferred by Eve.

**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 state, and hence can be regarded as one node. Therefore, we can only consider the case that 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 $\Delta x_{EA} = a_A^{(k)} a_E^{(k)} (x_A[k] - x_E[k])$. Similar to the proof of
After $K$ steps, the measurements collected by Eve are given by
\[ y_E[0 : K] = O_{E,[0 : K]} x[0] \]
with the observability matrix $O_{E,[0 : K]}$ having the same form as $O_A$.

Now in the $K + 1$ equations collected by Eve in (26), there are $K + 2$ unknowns $x_A[0], a_A^0, a_A^1, \ldots, a_A^K$. However, after converging to average consensus, Eve will be able to know the final state of other nodes (the same as its final state), which enables it to construct another equation about the initial states like (24). This will make the total number of equations equal to the total number of involved unknowns and make solving initial state of $x_A$ possible.

Next we use an example to illustrate that it is indeed possible for Eve to infer the state of Alice if Eve is Alice’s only neighbor. Consider the configuration in Fig. 2 (b). Eve receives $\Delta x_{E,A}[k] = -\Delta x_{E,A}[k]$ from Alice. In addition, after the protocol converges after $K$ steps, Eve knows the final state which is identical for all the nodes. The initial value of Alice can be simply inferred by Eve through
\[ x_A[0] = x_A[K] + \varepsilon \sum_{k=0}^{K-1} \Delta x_{E,A}[k] = x_E[K] - \varepsilon \sum_{k=0}^{K-1} \Delta x_{E,A}[k] \]
Therefore this single connection configuration should always be avoided, which is also required by other noise-based privacy protocols, for instance in [6] and [9].

**B. Security Solution**

Due to the additive homomorphic property, the Paillier cryptosystem is vulnerable to active adversaries who are able 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 $k_{p,A}$ is sent along with $E(-x_1)$, Eve may use the additive homomorphism to inject an arbitrary noise $\xi$ to the original message $E(-x_1)$ to sway it to $E(-x_1 + \xi)$. If Bob has no way to tell if 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 incoming message. We propose to attach a digital signature to the exchanged message in the confidential interaction protocol, based on which the recipient can verify possible modifications during communication. The signature requires an additional pair of public/private keys ($k_{p,A}'$, $k_{s,A}'$) and a hash function $H(\cdot)$, and is represented as $(k_{s,A}', E_A'[H(m)])$. The additional private key $k_{s,A}'$ is sent so that Bob can encrypt $E_A'[H(m)]$ and check if the resulting $H(m)$ matches the received $m$ in terms of the hash operation $H(\cdot)$ (cf. Fig. 3 (b)). Because without the public key $k_{p,A}'$, Eve cannot forge a valid signature (that can be decrypted by Bob), any Eve’s attempt to modify $m$ will cause a mismatch between received $m$ and decrypted $H(m)$ in terms of the hash operation $H(\cdot)$.

**VI. EXTENSION 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 all all states to a weighted sum of the initial state, i.e.,
\[ \sum_{j=1}^{n} w_j x_j(0) / \sum_{j=1}^{n} w_j \] with $w_j > 0$ being the weights. According to (11), weighted average consensus can be achieved by using the following update rules:

For continuous-time domain, the update rule is
\[ x_i(t) = \frac{1}{w_i} \sum_{j \in N_i} \Delta x_{ij}(t) \]
For discrete-time domain, the update rule is
\[ x_i[k + 1] = x_i[k] + \frac{\varepsilon}{w_i} \sum_{j \in N_i} \Delta x_{ij}[k] \]

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:** Under the confidential interaction protocol in Sec. III and any positive $a$, update rule (28) can achieve weighted average consensus in the continuous-time domain, i.e.,
\[ \lim_{t \to \infty} x_i(t) = \frac{\sum_{j=1}^{n} w_j x_j(0)}{\sum_{j=1}^{n} w_j}, \quad i = 1, 2, \ldots, n. \]
Proof: The proof can be obtained by following a similar line of reasoning of Corollary 3 in [28]. ■

Theorem 6: Under the confidential interaction protocol in Sec. III and any \( \bar{a} < 1 \), update rule (29) can achieve weighted average consensus in the discrete-time domain, i.e.,
\[
\lim_{k \to \infty} x_i[k] = \frac{\sum_{j=1}^{n} u_{ij} x_j[0]}{\sum_{j=1}^{n} u_{ij}}, \quad i = 1, 2, \ldots, n. \tag{31}
\]

Proof: The proof can be obtained by following a similar line of reasoning of Theorem 1 in [29]. ■

B. Maximum Consensus

The maximum consensus seeks the convergence of all states to the maximal initial value among all states, i.e.,
\[
\max_{j \in V} \{ x_j(0) \} = \max \{ x_j(0) \}. \quad \text{Next we show how to use the confidential interaction protocol in Sec. III to guarantee the privacy in maximum consensus. In contrast to the conventional maximum consensus protocol where the current state of a node is directly replaced with the maximum state among its neighbors [13], we formulate maximum consensus problem as a nonlinear dynamical system to fit into our confidential interaction protocol.}
\]

For continuous time, we design the update rule as
\[
\dot{x}_i(t) = \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_i(t)) \tag{32}
\]
For discrete time, we design the update rule as
\[
x_i[k + 1] = x_i[k] + \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j[k] - x_i[k]) \tag{33}
\]

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. The changes in discrete time case are analogous to that of the continuous time case except that we no longer need the step size \( \varepsilon \).

In this new maximum consensus approach, the weighted state difference can still be calculated using the confidential interaction protocol in Sec. III, the operation of multiplying a node’s own random weight to the weighted difference, i.e., \( a_1 \) in (13), can be removed to simplify computation because there is no need to guarantee a symmetric interaction graph any more. For this reason, here we always set \( a_1 \) in (13) to 1, which leads to the coupling weight in (32) and (33) above.

Theorem 7: Under the confidential interaction protocol in Sec. III and any positive \( \bar{a} \), update rule (32) can achieve maximum consensus in the continuous-time domain, i.e.,
\[
\lim_{t \to \infty} x_i(t) = \max_{1 \leq j \leq n} x_j(0), \quad i = 1, 2, \ldots, n. \tag{34}
\]

Proof: In order to prove Theorem 7, we follow three steps below. First, we show that the value of each node under the update rule (32) is non-decreasing. Then we show that the maximum value is time-invariant. Lastly, we prove that all values converge to the same value, which can only be the maximum among all initial state values.

First, we show that the value of each node under the update rule (32) is non-decreasing, i.e.,
\[
\frac{d}{dt} x_i(t) \geq 0, \quad \forall i, t. \tag{35}
\]
According to (32), we have
\[
\frac{d}{dt} x_i(t) = \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_i(t)) \\
\geq a_i(x_i(t) - x_i(t)) \\
= 0
\]
so \( \frac{d}{dt} x_i(t) \geq 0 \) always holds for \( i = 1, 2, \ldots, n \).

Next we show that the maximum value is invariant, i.e.,
\[
\max_{1 \leq i \leq n} \{ x_i(t) \} = \max_{1 \leq i \leq n} \{ x_i(0) \} \tag{37}
\]

Represent the maximum value among all the states as \( x_k(t) = \max_{1 \leq i \leq n} \{ x_i(t) \} \). By definition, we have
\[
\frac{d}{dt} x_k(t) = \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_k(t)) \\
\leq \max_{v_j \in V} a_j(x_j(t) - x_k(t)) \\
\leq \max_{1 \leq j \leq n} a_j(x_j(t) - x_k(t))
\]

Since \( a_j > 0 \) and \( x_j(t) - x_k(t) \leq 0 \) hold for \( j = 1, 2, \ldots, n \), we have \( \max_{1 \leq j \leq n} a_j(x_j(t) - x_k(t)) = 0 \), i.e., \( \frac{d}{dt} x_k(t) \leq 0 \).

Further taking into account the fact that the value of each node is non-decreasing, we have \( \frac{d}{dt} x_k(t) = 0 \). Note that no other \( x_i(t) \) will be larger than \( x_k(t) \) because \( x_k(t) \) is Lipschitz continuous for \( i = 1, 2, \ldots, n \) and \( \frac{d}{dt} x_k(t) = 0 \) holds once \( x_i(t) \) reaches \( x_k(t) \). Therefore, \( \max_{1 \leq i \leq n} \{ x_i(t) \} = \max_{1 \leq i \leq n} \{ x_i(0) \} = \alpha \).

Lastly, we show that all states converge to the maximum, i.e., \( \alpha \). We define the error vector as \( \delta(t) = x(t) - \alpha 1 \), where \( x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T \). Theorem 7 is thus equivalent to proving
\[
\lim_{t \to \infty} \delta(t) = 0 \tag{39}
\]
According to (32), the dynamics of \( \delta(t) \) is given by
\[
\frac{d}{dt} \delta(t) = -\max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_i(t)) \tag{40}
\]

Define the Lyapunov function \( V(\delta) = \frac{1}{2} \delta^T \delta \geq 0 \), which is zero only when \( \delta \) is zero, then we have
\[
\frac{d}{dt} V(\delta) = \dot{\delta}^T \delta \tag{41}
\]
\[
= -\sum_{v_j \in V} \left( \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_i(t)) \right) \left( \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_i(t)) \right)
\]
Since \( \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_i(t)) \leq 0 \) and \( \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} a_j(x_j(t) - x_i(t)) \geq 0 \) hold for \( i = 1, 2, \ldots, n \), we have
\[
\frac{d}{dt} V(\delta) \leq 0 \tag{42}
\]
Obviously, in the above inequality, equality holds when 
\[ \max_{v_j \in N_i \cup \{v_i\}} (x_j(t) - x_i(t)) = 0 \] 

or \[ \max_{v_j \in N_i \cup \{v_i\}} (x_j(t) - x_i(t)) = 0 \] holds for \( i = 1, 2, \ldots, n \), i.e., when the maximum consensus is achieved. So the Lyapunov function \( V(\delta) \) will keep decreasing until reaching zero, meaning that maximum consensus is achieved.

Theorem 8: Under the confidential interaction protocol in Sec. III and any \( 0 < \bar{a} < 1 \), update rule (33) can achieve maximum consensus in the discrete-time domain, i.e.,

\[ \lim_{k \to \infty} x_i[k] = \max_{1 \leq j \leq n} x_j[0], \quad i = 1, 2, \ldots, n. \]  

(43)

**Proof:** The proof follows the same structure as the proof of Theorem 7. First we prove that the value of each node under the update rule (33) is non-decreasing, i.e.,

\[ x_i[k + 1] - x_i[k] \geq 0, \quad \forall i, k \]  

(44)

According to (33), we have

\[ x_i[k + 1] - x_i[k] = \max_{v_j \in N_i \cup \{v_i\}} a_j (x_j[k] - x_i[k]) \]

\[ \geq a_i (x_i[k] - x_i[k]) \]

\[ = 0 \]  

(45)

So \( x_i[k + 1] - x_i[k] \geq 0 \) always holds for \( i = 1, 2, \ldots, n \).

Now we show that the maximum value is time-invariant under the update rule, i.e.,

\[ \max_{1 \leq i \leq n} \{x_i[k]\} = \max_{1 \leq i \leq n} \{x_i[0]\}, \quad \forall k \]  

(46)

Let \( x_p[k] = \max_{1 \leq i \leq n} \{x_i[k]\} \) at step \( k \). According to (33), we have

\[ x_p[k + 1] - x_p[k] = \max_{v_j \in N_i \cup \{v_i\}} a_j (x_j[k] - x_p[k]) \]

\[ \leq \max_{v_j \in V} a_j (x_j[k] - x_p[k]) \]

\[ \leq \max_{v_j \in V} (x_j[k] - x_p[k]) \]

\[ = \max_{v_j \in V} \{x_j[k]\} - x_p[k] \]

\[ = 0 \]  

(47)

In the derivation, we used the fact \( 0 < a_j < 1 \). Since the value of each node is non-decreasing, we have \( x_p[k + 1] = x_p[k] \).

So \( x_p[k] \) is time-invariant.

Next we show that all other nodes will stay less or equal to \( x_p[k] \). For every node \( i \), based on the fact \( x_i[k] \leq x_p[k] \), we have

\[ x_i[k + 1] = x_i[k] + \max_{v_j \in N_i \cup \{v_i\}} a_j (x_j[k] - x_i[k]) \]

\[ \leq x_i[k] + \max_{v_j \in V} a_j (x_j[k] - x_i[k]) \]

\[ \leq x_i[k] + \max_{v_j \in V} (x_j[k] - x_i[k]) \]

\[ = x_i[k] + \max_{v_j \in V} \{x_j[k]\} - x_i[k] \]

\[ = x_p[k] \]

\[ = x_p[k + 1] \]  

(48)

where we used \( 0 < a_j < 1 \). As a result, the maximum is invariant with respect to \( k \).

Let \( \max_{x_i \in V} \{x_i[k]\} = \max_{v_i \in V} \{x_i[0]\} = \alpha \). We can now define the error vector as \( \delta[k] = \alpha \bar{a} - x[k] \), where \( x[k] = [x_1[k], x_2[k], \ldots, x_n[k]]^T \). Note that \( \delta[k] \geq 0 \) holds for all \( i \) and \( k \). Theorem 8 is thus equivalent to

\[ \lim_{k \to \infty} \delta[k] = 0 \]  

(49)

According to (33), the dynamics of \( \delta[k] \) is given by

\[ \delta[k + 1] - \delta[k] = - \max_{v_i \in N_i \cup \{v_i\}} a_j (x_j[k] - x_i[k]) \]  

(50)

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

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

(51)

Expanding the right hand side yields:

\[ RHS = (\delta[k + 1] - \delta[k])^T (\delta[k + 1] + \delta[k]) \]

\[ = - \sum_{v_i \in V} \max_{v_i \in N_i \cup \{v_i\}} a_j (x_j[k] - x_i[k]) \cdot (\delta_i[k + 1] + \delta_i[k]) \]

\[ \leq 0 \]  

(52)

The equality holds when \( \max_{v_i \in N_i \cup \{v_i\}} a_j (x_j[k] - x_i[k]) = 0 \) holds for \( i = 1, 2, \ldots, n \), i.e., when the maximum consensus is achieved. So the Lyapunov function \( V(\delta[k]) \) will keep decreasing until the maximum consensus is achieved.

Theorem 9: Under the confidential interaction protocol in Sec. III and any positive \( \bar{a} \), update rule (33) can achieve minimum consensus in the continuous-time domain, i.e.,

\[ \lim_{t \to \infty} x_i(t) = \min_{1 \leq j \leq n} x_j(0), \quad i = 1, 2, \ldots, n. \]  

(55)

**Proof:** The proof follows the same line of reasoning as Theorem 7 and is omitted.

Similar to Theorem 8, we can obtain the following results for minimum consensus in the continuous-time domain:

Theorem 10: Under the confidential interaction protocol in Sec. III and any \( \bar{a} \) satisfying \( 0 < \bar{a} < 1 \), update rule (54) can achieve minimum consensus in the discrete-time domain, i.e.,

\[ \lim_{k \to \infty} x_i[k] = \min_{1 \leq j \leq n} x_j[0], \quad i = 1, 2, \ldots, n. \]  

(56)
Proof: The proof follows the same line of reasoning as Theorem 8 and is omitted.

VII. IMPLEMENTATION DETAILS

In addition to the constraints imposed on \( a_i \) and \( \varepsilon \), 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 \( f(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \to \mathcal{M} \subseteq \mathbb{Z} \) and its inverse \( f^{-1}(\cdot, \cdot) : \mathcal{M} \times \mathbb{R} \to \mathbb{R} \) as

\[
 f(x, N) = \lceil Nx \rceil \mathcal{M}, \quad f^{-1}(y, N) = \frac{y}{N}
\]

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 \( \Delta = 1 \) which is uniform. Consequently the maximum quantization error is bounded by

\[
 \max_{x \in \mathbb{R}} |x - f^{-1}(y, N)| = \frac{\Delta}{N}
\]

In practice we choose a sufficiently large value for \( N \) 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. [18] 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 EXAMPLES

To illustrate the capability of our protocol, we implemented the discrete-time version of the consensus protocol in C/C++. We used an open-source C implementation of the Paillier cryptosystem [30] because it allowed byte-level access. For each exchange between two nodes, the states were converted to 64-bit integers by multiplying \( N = 10^5 \). The weights \( a_i(t) \in (0,1) \) 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 and we set the step-size to \( \varepsilon = 0.5 \). The initial states were set to \( \{1.0, 2.0, 4.0, 8.0\} \) 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.

The second simulation considered the weighted average consensus. Given 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. 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 10 steps.

The computational overhead caused by the encryption is
resilience to passive attackers and allows easy incorporation of average in a deterministic manner. The protocol also provides cryptosystem which allows the convergence to the exact average in a deterministic manner. In contrast to previous approaches where the states are covered with random noise which unavoidably affects the convergence performance, we encode randomness to the convergence problem and they are not applicable to other types of consensus problems without significant modification.

On the other hand, although the computational overhead introduced by the encryption algorithm (7 ms) is still 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 manageable on modern computers.

**ACKNOWLEDGEMENT**

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

**REFERENCES**

[1] M. H. Degroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[2] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–223, 2007.

[4] J. N. Tsitsiklis. Problems in decentralized decision making and computation. PhD thesis, 1984.

[5] 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.

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

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

[10] E. Nozari, P. Tallapragada, and J. Cortés. Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design. *Automatica*, 81(7):221–231, 2017.

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

[12] 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.

[13] 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.

**IX. DISCUSSIONS AND CONCLUSIONS**

In this paper we proposed a decentralized secure and privacy-preserving protocol for the network consensus problem. In contrast to previous approaches where the states are covered with random noise which unavoidably affects the convergence performance, we encode randomness to the system dynamics with the help of an additive homomorphic cryptosystem which allows the convergence to the exact average in a deterministic manner. 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 a new update rule. Simulations results are provided to demonstrate the effectiveness of the approach.

Comparing to the noise based average consensus protocols by Huang et al. [8] and Mo and Murray [6], our approach offers several advantages. First our approach guarantees that the convergence to the average value is exact, whereas as shown in [8] Huang’s algorithm may converge to a different value other than the exact average.

Secondly, the results in [6] and [8] are based on time-invariant parameters, which could be troublesome if the topology of the network or the number of nodes is not constant. Consider the scenario when a new node is added to a network after the network has converged. The the arrival of the new node must be broadcasted to all existing nodes to reset their noise parameters or they risk exposing their current states.

Last but not least, the two aforementioned noise-based protocols are specifically tailored for the average consensus problem and they are not applicable to other types of consensus problems without significant modification.

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.
[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] R. L. Lagendijk, Z. Erkin, and M. Barni. Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multi-party computation. IEEE Signal Processing Magazine, 30(1):82–105, Jan 2013.

[21] 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.

[22] 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.

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

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

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

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

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

[28] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on Automatic Control, 49(9):1520–1533, 2004.

[29] 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.

[30] J. Bethencourt. Advanced crypto software collection. [http://asc.cs.utexas.edu/libpailler](http://asc.cs.utexas.edu/libpailler)