Thesis

Metin Çalış

September 4, 2019

Privacy-Preserving Consensus Averaging

by

Metin Çalış

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4738675	
Project duration:	January 9, 2019 – September 20, 2019	
Thesis committee:	dr. ir. R. Heusdens,	TU Delft, supervisor
	dr. ir. R. C. Hendriks,	TU Delft, supervisor
	dr. Z. Erkin,	TU Delft,

TU Delft

Acknowledgements

I would like to start by thanking to my supervisors Richard Heusdens and Richard Hendriks for their valuable remarks and guidance which provided me the confidence and determination to finish this thesis. My family, Esin, Vural, Melih, Hare Nil Çalış and Nesrin Göktuğ, for their absolute trust and care they have provided throughout my studies. Finally my friends, scattered across the world pursuing degrees of their own, who have found time to spare a piece of their mind whenever I asked.

Abstract

Consensus problem has been a topic of interest for many research areas allowing multiple agents to reach an agreement through local information exchange. The explicit share of the state variables, however, may cause privacy issues due to the confidentiality of the initial values. In this work, asynchronous privacy-preserving consensus average algorithms are proposed which enables the agents to reach the exact average of their initial values while preserving the privacy of them. The research aims to reduce the convergence time and computational complexity compared to the cryptographic solutions. Three methods are proposed and compared. The state decomposition and noise-obfuscation methods preserve the privacy of the initial values given that the semi-honest adversary is not able to listen to one of the neighboring nodes of the targeted node. The hybrid state decomposition approach proposes a way to overcome this assumption by using a minimum number of encryption operations. The initial values are shown to be private against an eavesdropper who is able to tap all communication channels as well as a semi-honest adversary in the system. It has been shown in all proposed approaches that as the noise variance goes to infinity, the adversary does not have any range to estimate the initial value. The noise obfuscation technique futures the same convergence rate as the standard averaging approach while providing a linear increase in the variance of the adversary's estimate with the increasing noise variance. On the other hand, the state decomposition technique futures a lower convergence rate compared to the standard averaging approach while providing an exponential increase in the variance of the adversary's estimate with the increasing noise variance. By optimizing the algorithm, it has been shown that the same convergence rate as the standard randomized gossip can be obtained. The state decomposition approach requires the addition of noise for a bounded amount whereas, the noise obfuscation method requires the addition of noise at each iteration. All three approaches, converges faster than a fully cryptographic approach while promising statistical security guarantees.

Keywords: Distributed average consensus, privacy-preserving, noise-obfuscation, state decomposition, asynchronous average consensus.

Technical report:

Signal and Information Processing Lab
Delft University of Technology
2628 CD Delft, The Netherlands

Contents

List of Figures

Chapter 1

Introduction

Consensus problem in dynamic systems has been a topic of interest that has found usage in many research areas allowing multiple agents to reach an agreement through local information exchange between the agent and its neighbors[1]. Some of these research areas are sensor fusion [2][3][4], control of swarms and flocks [5][6], alignment problem [7] and asynchronous consensus [8]. In the consensus averaging problem, N nodes in an undirected graph reach to the average of their initial values,

$$\frac{1}{N}\sum_{i=1}^{N} x_i[0]$$

where x represent the nodes state variables and i represent the ith node. There has been many distributed algorithms such as gossip algorithms [9], alternating direction method of multipliers(ADMM)[10], augmented Lagrangian methods (ALM) [11], primal-dual method of multipliers (PDMM)[12] that were proposed to solve the averaging problem in a distributed manner.

The traditional consensus algorithms explicitly exchange their state variables to solve a common function. However, for some consensus problems such as the multi-rendezvous problem [13] or energy management in smart grids [14], the initial states can be confidential. In the former, the agents might not want to reveal their initial locations while in the latter, the energy companies might not want to reveal their individual generation rates. The challenge to solve the consensus problem while giving individual nodes a privacy guarantee initiated the new privacy-preserving distributed optimization research area.

The privacy addressed to the nodes can be in a few different forms. In the multi-rendezvous problem, the initial states are considered to be the secret. Whereas, in projection based source localization problems, the intermediate states can be considered as the secret as three intermediate states would help any node to infer the position of the neighboring node which could be undesirable[15][16]. In the regression problem[17], the objective functions and the gradients can be considered confidential since they can include private information such as salary. In the differential privacy based approaches[18], the state variables and the final average are protected from the adversaries[19]. This line of work provides formal confidentiality guarantees using the privacy notion proposed for statistical databases. According to the problem at hand, the confidentiality requirement is defined and a respective algorithm is designed.

In this thesis, the averaging problem is considered with an emphasis on the privacy of the initial values. The second chapter introduces the related work regarding the privacy-preserving distributed optimization and formulates the research question that complies with the state-of-art. The third chapter gives the preliminary information that is used in the following chapters; the distributed averaging problem, the implemented gossip algorithm, privacy and the used tools are explained in detail. The following three chapters future the algorithms which are proposed to solve the privacy-preserving asynchronous distributed averaging problem. The fourth chapter is about the privacy-preserving state decomposition algorithm which transforms the consensus averaging problem into a constrained multi-agent distributed problem[20] and solves it instead. The fifth chapter is an extension of the state decomposition algorithm where the scope of the privacy guarantee is improved. The sixth chapter presents how adding correlated and decaying noise to the state variables can solve the investigated problem. Both of the three chapters include the same format which covers the methodology, privacy and numerical analysis attributes. The final chapter concludes with a comparison of the proposed algorithms and recommendations for future work.

Chapter 2

Related Work

The research directed towards solving the consensus averaging problem while preserving the privacy of initial values can be categorized into two approaches: cryptographic[21][22] and non-cryptographic methods[23][24][25][26]. Most of the cryptographic methods use homomorphic encryption to encrypt the states that are being transferred. Due to the redundancy and randomness introduced in the ciphertext, the cryptographic algorithms provide high dimensional security at the expense of computational complexity. Another technique is to use garbled circuits[27] to implement the consensus function using a privacy-preserving manner. Although many optimization techniques[28][29] are proposed, the garbled circuits are also computationally complex and slow compared to their non-cryptographic counterparts.

In control and real-time dynamic systems where processing time is limited, or distributed solvers which solve optimization problems iteratively, cryptographic methods are not suitable due to the time the encryption and the decryption takes. To reduce the time and complexity, privacy-preserving non-cryptographic consensus methods are proposed. The methods to solve this problem can be categorized into three parts: differential privacy [23][19], noise-obfuscation [24][26] and transformation methods[25]. Differential privacy based approaches trade accuracy for privacy. The nodes add noise to the transmitted states and provide a differential privacy guarantee as defined in [18] or in [30] for continuous data observations. However, as proven by [23] differential privacy and exact consensus cannot be achieved simultaneously. The noise-obfuscation methods, on the other hand, add correlated noise to the transmitted states. As the added noise is zero-sum and decaying in magnitude, the exact average can be achieved. The privacy is analyzed by examining the covariance matrix of the maximum likelihood estimate [26] and extended to (ϵ, δ) privacy [24] where ϵ and δ represent the range and estimation confidence respectively. The transformation method [25] solves a constrained optimization problem as a variant of the averaging problem. The initial value of each node is decomposed into two states where one of them is used to interact with the other nodes and the other is used internally affecting the evaluation of the former indirectly. Through a one-shot perturbation of the states with additive noise, the authors achieve a synchronous privacy-preserving algorithm with the assumption that one of the update weight is hidden from the adversary.

Some networks are constrained by few characteristics which prevent an application of a synchronous algorithm. The network might lack a centralized entity which processes all the information and synchronizes the time, the power

resources and compute power may be limited or the network topology can be varying. For this reason, asynchronous distributed optimization techniques such as gossip algorithms [9] and convex optimization based algorithms [10][12] were proposed. To the best of our knowledge only [31] examines the privacy of initial values in a randomized gossip setting using noise-obfuscation techniques. The authors analyze the convergence conditions and rate of convergence without quantifying the provided privacy. In this thesis, the aim is to examine the asynchronous privacy-preserving averaging methods that are faster than cryptographic approaches. The provided privacy is analyzed and a comparison is provided. In summary, this thesis tackles the following challenges:

- The nodes will reach to the average of their initial values, $x_{\text{ave}} = \sum_{j=1}^{N} x_j[0]$ through asynchronous updates.
- Each nodes initial value will be hidden from the adversaries throughout the process.
- The convergence time and computational complexity should be reduced compared to cryptographic solutions.

Chapter 3

Background Information

3.1 Distributed Averaging Problem

The undirected graph G is represented as $G = (V, E)$ with nodes being represented as $V = \{v_1, v_2, ..., v_N\}$ and the edge set as $E \subset V \times V$. The ith component of the vector $x[k] = [x_1[k], x_2[k], ..., x_N[k]]$ represent the state of node v_i at the iteration k. The set of neighbors of node v_i is $N_i = \{v_j \in V : (v_i, v_j) \in E\}$ and its cardinality is shown as $|N_i|$. The nodes aim to reach to the average of their initial values, $x_{\text{ave}} = \sum_{j=1}^{N} x_j[0]$, through communicating only with their neighbors.

There have been several methods proposed to do distributed consensus averaging such as distributed linear averaging algorithms [32], gossip algorithms [9] and convex optimization based algorithms [10][12]. According to the number of the node states updated at each round, the averaging methods can be categorized into two: synchronous and asynchronous algorithms. The distributed linear algorithms are synchronous algorithms, the gossip algorithms are asynchronous whereas, the convex optimization based methods can be both.

In synchronous distributed averaging algorithms, all the nodes update their state variable via a weighted average of their neighbors' state variables at each iteration. The synchronous updates require a shared clock and are sensitive to changes in topology. However, synchronous algorithms promise convergence to the average in fewer iterations compared to the asynchronous algorithms. In asynchronous algorithms, only two nodes update their states at each iteration while the rest keep their state variables for the next iteration. The asynchronous algorithms do not require a shared clock and more robust against changes in the network topology.

3.2 Randomized Gossip

Some networks are constrained by few characteristics which prevent an application of a synchronous algorithm. The network might lack a centralized entity which processes all the information and synchronizes the time, the power resources and the computing power may be limited or the network topology can be varying. Gossip algorithms[9] are proposed to overcome this problem. In gossip algorithms, only two nodes update their state variables at a time.

This reduces the computational burden that would otherwise be on the central node which communicates with all its neighbors. There are several gossip algorithms such as randomized gossip, geographic gossip, greedy gossip with eavesdropping and weighted gossip. The update of randomized gossip algorithm is given by the following.

$$x(k) = W(k)x(k-1)$$

As the iterations continue, it is expected for the $x(k)$ to converge to the average of the initial values. The structure of the $W(k)$ can be given as

$$\lim_{k\to\infty} \phi(k) = \lim_{k\to\infty} W(k)\dots W(1) = \frac{\mathbf{1}\mathbf{1}^T}{n}.$$

Let the expected value of the matrix be denoted as $E\{\phi(k)\}$ be

$$E\{\phi(k)\} = \prod_{i=1}^{k} E\{W(k)\} = \bar{W}^k$$

such that the iterations converge to $\frac{\mathbf{1}\mathbf{1}^T}{n}$ if $\bar{W}^k$ converges to $\frac{\mathbf{1}\mathbf{1}^T}{n}$. Three conditions[9] are given for the system to converge. These are

$$\mathbf{1}^T\bar{W} = \mathbf{1}^T$$
$$\bar{W}\mathbf{1} = \mathbf{1}$$
$$\rho(\bar{W} - \frac{\mathbf{1}\mathbf{1}^T}{n}) < 1$$

Let the weight matrix at the iteration k to be selected with probability P_{ij}/N as

$$W_{ij} = I - \frac{1}{2}(e_i - e_j)(e_i - e_j)^T$$

where P_{ij} is the probability that node v_i contacts node v_j, e_i is the i standard basis vector in $\mathbb{R}^n$. The expected value of the matrix $\bar{W} = E\{W(k)\}$ is

$$\bar{W} = \frac{1}{N}\sum_{i,j\in E} P_{ij}W_{ij}$$

It can be seen that $\bar{W}$ is doubly stochastic, i.e. the rows and the columns all add up to 1. Doubly stochasticity suggests that $(1, \mathbf{1})$ is an eigenpair of the matrix and $|\lambda(\bar{W})| \leq 1$. Since W_{ij} is a projection matrix, $W_{ij}^k = W_{ij}$, and a positive semi definite matrix, it can be deduced that $\lambda(\bar{W}) \in [0, 1]$. $\bar{W}$ is non-negative, irreducible and has a nonzero diagonal element. Perron-Frobenius theorem suggests one is a simple eigenvalue of the matrix and the rest are strictly less than zero. Thus, $\rho(\bar{W} - \frac{\mathbf{1}\mathbf{1}^T}{n}) < 1$ and the randomized gossip will converge to the average of the initial values.

3.3 Privacy

The notion of preserving privacy in the secure signal processing domain informally means the privacy of the data and the privacy of the algorithm[33]. Throughout the literature, the privacy of the data is more commonly investigated

although there can be situations where the service provider is also concerned about the privacy of the algorithm. At many of the secure signal processing algorithms, there are interactive protocols which leak knowledge about the underlying functions. In that case, the choice of secure signal processing schemes which enable more homomorphic operations on the encrypted data becomes more reasonable.

The privacy of the data can be achieved using two different methods: a trusted third party and secure signal processing. For the case of trusted third parties, the service provider takes the input from the parties and the algorithm from the service provider to compute the function and return the result to the parties. The inclusion of a third party is found to be risky and expensive[33]. Secure signal processing, on the other hand, promotes the involved parties to work directly on the data.

According to the application type, the definition of privacy of the data alters. The privacy of the data may not be only the objective function but also the intermediate states and subgradients[34]. Confidentiality of these values can be as important as the objective function as they are dependent on each other. The intermediate states may expose sensitive information about the hidden state variables. Throughout this thesis study, the privacy of the data has been examined. The main goal of the nodes is to reach to the consensus while hiding initial values from the other nodes throughout the execution. The methods that promote the privacy of the initial values can be broadly classified into two categories: cryptographic approaches and non-cryptographic approaches[35].

3.3.1 Non-cryptographic Approaches

The non-cryptographic approaches have lower complexity and computational overhead compared to cryptographic approaches. There is no formal definition of privacy agreed by the research community for non-cryptographic approaches. Some authors observe the covariance of a general unbiased estimator of the initial values to prove privacy[26], some show the indistinguishability of the information output of the adversary under an arbitrary change of the initial values[25][36] whereas, some use a privacy definition called $\epsilon-$ differential privacy[23][19].

$\epsilon-$differential privacy

Differential privacy is a statistical technique which prevents uncovering of individual records out from a database through means of random perturbations[35][18]. How much noise is added is dependent upon how private the algorithm is designed to be. The magnitude of the noise depends on the largest change a single entry can do on the output. This is defined as *sensitivity* of the algorithm and noise is selected such that ϵ differential privacy is guaranteed. This way of hiding data, however, trades off accuracy for privacy which leads to problems at the domains where small errors are meaningful such as medical diagnosis.

The concept of differential privacy on average consensus problem has been proposed by Huang et. al.[19] which lays its foundation on the concept of differential privacy for continuous bit streams [30]. The work has been extended to have an asymptotically unbiased algorithm which has an almost sure convergence guarantee and an explicitly characterized convergence rate. In this method, the agents add non-zero sum and decaying Laplacian noise to the

messages that they transmit. As the iterations continue, the noise becomes considerably small which enables nodes to converge.

The differential privacy[23] is defined as the following. Given $\delta \in \mathbb{R}_{>0}$, the initial network states $\mathbf{x}^1[0]$ and $\mathbf{x}^2[0]$ are $\delta - adjacent$ if for some node $v_m \in N$,

$$|x_i^2[0] - x_i^1[0]| \leq \begin{cases} \delta & , \text{if } i = m \\ 0 & , \text{if } .i \neq m \end{cases}$$

for all nodes $v_i \in N$. Given $\delta, \epsilon \in \mathbb{R}_{\geq 0}$, an algorithm is said to be $\epsilon-$differentially private if, for any pair $\mathbf{x}^{(1)}[0]$ and $\mathbf{x}^{(2)}[0]$ of δ adhjacent initial states and any set $\mathcal{O} \in \boldsymbol{B}((\mathbb{R}^N)^{\mathrm{K}}$,

$$\mathrm{P}\{\eta \in \Omega | X_{\mathbf{x}^{(1)}[0]}(\eta) \in \mathcal{O}\} \leq e^{\epsilon} P\{\eta \in \Omega | X_{\mathbf{x}^{(2)}[0]}(\eta) \in \mathcal{O}\}\,.$$

where $\boldsymbol{B}$ represent the borel set, K represents the total iteration number, $X_{\mathbf{x}^{(2)}[0]} : (\mathbb{R}^N)^{\mathrm{K}} \rightarrow (\mathbb{R}^N)^{\mathrm{K}}$ represent the function that takes an initial value set $\mathbf{x}[0]$ and a noise set $\eta \in (\mathbb{R}^N)^{\mathrm{K}}$ which is the horizontally stacked noise samples for K iterations and returns the horizontally stacked state variables $\mathbf{x} \in (\mathbb{R}^N)^{\mathrm{K}}$. In non-formal words, δ adjacent initial values represent two initial value vectors whose one element is different by δ and the rest the same. The ϵ differential privacy mean for δ adjacent initial states, the $N \times K$ vector of state variables that occurs by to the addition of the noise set will have a similar probability of existing among the borel set of $N \times K$ vectors. In other words, the possibility of observing a δ difference in the initial set is considerably small by observing the state variables that are released at each iteration.

3.3.2 Cryptoraphic Approaches

The data privacy of the cryptographic approaches is commonly investigated using cryptographic security definitions. There are two types of encryption schemes: the information-theoretic and the complexity-based. The security of these schemes is modeled using ideal world adversaries which imitate the attacks on the real world. If the system is proven to be secure only with the hard-problem assumption, standard oracle model (SOM) is used. Whereas, if involved primitives such as hash functions are assumed to be truly random in addition to the hard-problems, random oracle model is used.

Information-theoretic encryption

The first information-theoretic encryption scheme is suggested by Shannon [37]. The main premise of information-theoretic encryption schemes is that even a computationally unbounded adversary cannot break the ciphertext. The reason behind this is that the security of information-theoretic encryption schemes do not rely on hard problems like complexity-based versions. The information-theoretic secure encryption scheme (perfect secrecy/unconditional security) will have the property that the revelation of ciphertext does not leak any information about the message itself. For a message m and ciphertext $c = Enc(m)$, information-theoretic security means

$$p(m|c) = p(m).$$

To use this encryption scheme however, the key length should be as long as the message to be encrypted which makes this scheme practically infeasible to encrypt long messages. The other encryption scheme is complexity-based encryption schemes which rely on a problem that is assumed to be hard.

Complexity-based encryption schemes

The one-way trapdoor functions are functions that are easy to solve if the key is known however fairly hard to solve without the knowledge of the keys. Some examples of this are the factorization problem and the discrete log problem. The security of an encryption scheme is established if the cryptographic technique can be reduced in polynomial time to solving the problem that is assumed to be hard. In addition to this, just like advanced encryption standard(AES), the nature of the encryption can on itself be a hard problem.

The security of these techniques is examined through using adversaries with capabilities and attacks they can perform. There are many adversaries but the most commonly used ones are the semi-honest, covert and malicious adversary. The properties of adversaries are: computationally bounded and unbounded, static or adaptive where the behaviour of adversary can change any time, eavesdropping where passive listening occurs or Byzantine where active manipulation occurs[38].

- Semi-Honest Adversary: In the semi-honest adversarial model, the adversary doesn't manipulate the protocol or the data but rather tries to get information by observing and keeping the records of the data.
- Covert Adversary: A protocol is said to be ϵ deterrent safe against a covert adversary if for any party that is cheating can be caught with probability ϵ.
- Malicious Adversary: The malicious adversary can modify the protocol to gain more information. Any party that is participating in the protocol can change their inputs and outputs any time they want.

The level of security under any given adversary is defined by the possible attacks that the adversaries may perform. The most basic one among the semantic securities is the indistinguishable passive attack(IND-PASS). The level of security increases as the capabilities of attacks that the adversaries may do. Starting from the lowest security level to the highest, the cryptographic security definitions are as follows.

- Indistinguishablity under passive attack(IND-PASS): There is a key generated and a selection bit b which is used to define the messages, m_b. There are two messages $m_0, m_1 \in m_b$ where the challenger encrypts one and gives it to the adversary. If the advantage of the adversary is zero, i.e. the only thing that she can do is guess b with 0.5 probability, it is said that the encryption is IND-PASS secure.
- Indistinguishablity under chosen plaintext attack(IND-CPA): The same key and bit selection like IND-PASS is designed. However, now the adversary has the encryption oracle where she can ask the encryption of any message. Under this assumption, if she can guess b with 0.5 probability, it is said to be IND-CPA secure.
- Indistinguishablity under non adaptive ciphertext attack(IND-CCA1): The same key and bit selection like IND-PASS is designed. The adversary has the encryption oracle and the decryption oracle. Different to its IND-

CCA2 variant, in this one, the adversary can use both of the oracles arbitrary times before getting the challenge ciphertext. Under this assumption, if she can guess b with 0.5 probability, it is said to be IND-CCA1 secure.

- Indistinguishablity under adaptive ciphertext attack(IND-CCA2): The only difference between IND-CCA2 and IND-CCA1 is that the adversary can continue to use both encryption and decryption oracles after getting the challenge ciphertext. For the decryption oracle, she can ask for any message but the challenge ciphertext.

For the semi-honest adversary, the IND-CPA encryption scheme will suffice however for the malicious adversary, IND-CCA secure schemes are needed. This is due to the requirement of authentication schemes which prevent the manipulation of the data. According to the application domain and whether the parties are trusted to follow the procedure, the adversarial type will be defined. According to the adversarial type, an encryption scheme that satisfies the security requirement will be selected.

3.4 Additively Homomorphic Encryption

Homomorphic encryption enables the parties to operate on encrypted messages. There are four possible types of homomorphism. First one is additively homomorphic, second is multiplicatively homomorphic, the third one is somewhat homomorphic and the last one is fully homomorphic encryption. Each type defines the operations that can be done on the ciphertexts.

In the additively homomorphic encryption, multiplication of two ciphertexts results in the addition of the underlying texts given that they are generated using the same public key.

$$D_{sk}(E_{pk}(m_1) \times E_{pk}(m_2)) = m_1 + m_2 \tag{3.1}$$

In addition to this if the multiplicative inverse of the second message is taken, the result is the subtraction of two messages.

$$D_{sk}(E_{pk}(m_1) \times E_{pk}(m_2)^{-1}) = m_1 - m_2$$

It can also be seen that any ciphertext raised to the power of k results in the encryption of $k \cdot m$.

$$D_{sk}(E_{pk}(m)^k) = k.m$$

Paillier encryption [39] is one of the most commonly used additively homomorphic encryption scheme in multi-party computation(MPC) algorithms[40][41]. The generation of the keys, encryption and the decryption algorithms can be seen below.

- Key generation: the key owner computes $N = pq$ and $\lambda = lcm(p-1, q-1)$ and selects a random integer $g \in Z^*_{N^2}$ such that N divides the order of g. The public encryption key is (N, g) and the private decryption key is λ.

- Encryption:the encryption of the message $m \in Z_N$ is c with

$$c = g^m r^N \ mod \ N^2,$$

where $r \in_R Z_N$.

- Decryption: given the encryption $c \in Z_N^2$, underlying text can be obtained by

$$m = L(c^\lambda \ mod \ N^2)\mu \ mod \ N \text{where } \mu = (L(g^\lambda \ mod \ N^2))^{-1} \ mod \ N \text{and } L(u) = \frac{u-1}{N}$$

The messages lie in the domain of N whereas the ciphertexts lie in the domain of N^2. N is chosen as the multiplication of two large prime numbers such as 1024 bit p and q. Thus with a choice of large N, the constraint that the message needs to lie in the domain becomes feasible. Randomization is done by the random value r which destroys the deterministic nature of the encrypting same messages to the same ciphertext. The same r is not needed in the decryption which makes Paillier encryption an IND-CPA secure encryption scheme. The homomorphic property of this scheme can be shown using two messages m_1 and m_2 along with two random values r_1 and r_2

$$\begin{aligned} E_{pk}(m_1, r_1).E_{pk}(m_2, r_2) &= g^{m_1}.r_1^n.g^{m_2}.r_2^n \ mod \ n^2, \\ &= g^{m_1+m_2}.(r_1.r_2)^n \ mod \ n^2, \\ &= E_{pk}(m_1 + m_2, r_1.r_2). \end{aligned}$$

Homomorphic encryption cannot be IND-CCA2 secure due to the nature of possible operations that can be done to obtain the challenge ciphertext. The best security that can be achieved is IND-CCA1 security[42]. Through using possible transformations such as Fujiyaka-Okomoto transform[43] or limiting the users who can do homomorphic operations[44], it is possible to achieve IND-CCA secure schemes from IND-CPA secure encryptions. According to the application type, the security level and the computational overhead is traded.

3.5 Non Parametric Mutual Information Estimator

In information theory, mutual information has been a measure of independence and it plays an important role to understand the amount of information shared between the random variables. There have been several methods proposed to do non-parametric mutual information estimation such as binning and k-nearest neighbor estimation[45][46]. For the analysis of conditional mutual information, k-nearest neighbor mutual information estimator is used as it is shown to be a powerful estimator that is data-efficient and adaptive with minimal bias. Typically, there are $z_i = (x_i, y_i)$, $i = \{1, ..., N\}$ independent identically distributed (iid) realizations of a random variable Z=(X,Y) with density $\mu(x, y)$. Assuming that the integral exists and $0 \log 0 = 0$, the mutual information can be defined as,

$$I(X, Y) = \int \int dxdy\mu(x, y) \log \frac{\mu(x, y)}{\mu_{\mathbf{x}}(x)\mu_{\mathbf{y}}(x, y)}.$$

The shannon entropy in classical sense is defined as,

$$H(X) = -\int dx\mu(x) \log \mu(x)$$

The mutual information and conditional mutual information are based on the definition of Kozachenko-Leonenko estimate for the Shannon entropy[47]. A brief introduction to Kozachenko-Leonenko is as following. The entropy can be thought as the averages of $\log \mu(x)$ up to the minus sign. Thus, given an unbiased estimate of $\log \mu(x)_i$ the entropy can be written as,

$$\hat{H}(X) = -\frac{1}{N}\sum_{i=1}^{N} \log \hat{\mu}(x_i)$$

Let the distance from $z_i = (x_i, y_i)$ to another neighbour $z_j \neq z_i$ be represented as the following.

$$||z - z'|| = max||x - x'||, ||y - y'||,$$

where the distance operator is defined to be the maximum norm. Denote $\epsilon(i)/2$ to be the distance from z_i to its kth neighbour and $\epsilon_x(i)/2$ and $\epsilon_y(i)/2$ be the distances that are projected into the X and Y subspaces. To estimate the $\log \mu(x_i)$, the probability distribution $P_k(\epsilon)$ that represents the distance between x_i and kth nearest neighbor is considered. Probability $P_k(\epsilon)d\epsilon$ gives the chance that there is one other x_i in approximately at a $\epsilon/2$ distance, $k-1$ other points at smaller distances than $\epsilon/2$ and $N-k-1$ points at a larger distance than $\epsilon/2$. Denote $p_i(\epsilon) = \inf_{||\xi - x_i|| < \epsilon/2} d\xi$ the mass of the ϵ ball centered at x_i. Using the trinomial formula,

$$P_k(\epsilon)d\epsilon = k\binom{N-1}{k}\frac{dp_i(\epsilon)}{d\epsilon}p_i^{k-1}(1-p_i)^{N-k-1}.$$

The expected value of $\log p_i$ is then given by,

$$E(\log p_i) = \Psi(k) - \Psi(N),$$

where $\Psi(x)$ is the digamma function. Assuming that $\mu(x)$ is constant in the ϵ sphere, the entropy can be written as,

$$\hat{H}(X) = -\Psi(k) + \Psi(N) + \frac{d}{N}\sum_{i=1}^{N} \log \epsilon(i) \tag{3.2}$$

If this entropy definition is extended to the joint entropy cases, the only part that changes is dimension of the last term in eq. (3.2).

$$\hat{H}(X,Y) = -\Psi(k) + \Psi(N) + \frac{d_x + d_y}{N}\sum_{i=1}^{N} \log \epsilon(i) \tag{3.3}$$

The k-nearest distance for the joint entropy and the marginal entropy changes since given that same k is used, the distance in the joint entropy would always be greater than the distance in the marginal entropy. To overcome this issue [45] uses a varied k as eq. (3.2) holds for any k. The new updated equation is given by,

$$\hat{H}(X) = -\frac{-1}{N}\sum_{i=1}^{N} \Psi[n_x(i)+1] + \Psi(N) + \frac{d}{N}\sum_{i=1}^{N} \log \epsilon(i), \tag{3.4}$$

where $n_x(i)$ represents the number of points within the vertical lines $x = x_i \pm \epsilon(i)/2$. The same approach is used to estimate $H(Y)$ which is biased compared to $H(X)$. The reason is that $\epsilon(i)$ is not exactly equal to twice the distance

to the $(n_y(i)+1)$st neighbour. However, the authors claim that this is still a good approximation and this bias goes to zero as $N \to \infty$. The mutual information has the following relation with the entropy.

$$I(X;Y) = H(X) + H(Y) - H(X;Y) \tag{3.5}$$

If the estimates of eq. (3.2) and eq. (3.3) are plugged into eq. (3.5), the following estimate is obtained.

$$I(X;Y) = \Psi(k) + \Psi(N) - \Psi(n_x+1) - \Psi(n_y+1) \tag{3.6}$$

where $\Psi(x)$ is the digamma function. This estimate is shown in paper as $I^{(1)}(X,Y)$. Using the Kozachenko-Leonenko Estimate for Shannon entropy and a similar approach to using different k scales like [45], the authors of [46] found out that conditional mutual information $I(X;Y|Z)$ can be estimated as the following.

$$I(X;Y|Z) = \Psi(k) - \Psi(n_{xz}+1) - \Psi(n_{yz}+1) + \Psi(n_z+1) \tag{3.7}$$

where n_{yz} represents the k-nearest number of points in the $\epsilon(i)$ distance using the joint entropy $H(Y;Z)$. The toolbox [48] which implements eq. (3.7) is used during the information theoretic analysis of privacy.

Chapter 4

Privacy-Preserving Asynchronous Averaging using State Decomposition Method

4.1 Introduction

The aim of this algorithm is for N agents to reach to the exact average of their initial values while preserving the privacy of them through decomposing the initial values to two states called the alpha and beta states such that their sum is twice the initial value. Alpha states are used for communicating with the other nodes while beta states are used internally. Although the beta states are never shared in the system, they are used in the update of the alpha states at each iteration. Since the initial alpha states which will be released in the first iteration are selected randomly from the set of all real numbers, initial values will not be disclosed. The beta value, on the other hand, can be considered as the secret that relates the alpha states directly to the initial value. Since beta states are also included in the update of the alpha states, some information is leaked into the system about the initial values at each iteration. For this reason, a privacy analysis against the attacks of the semi-honest adversary and the eavesdropper is done that analyses this information leakage in addition to the analysis of the final privacy of the system. It has been found that if there is one hidden coupling weight between the target node and its neighbor, the adversaries do not have any range to estimate the initial values.

The state decomposition approach is proposed by [25] for the synchronous distributed averaging problem. In this chapter, the state decomposition approach is applied for the randomized gossip framework where the consensus is achieved asynchronously. The structure of the chapter is as follows. In section 4.2, the methodology of the state decomposition is explained via the initialization phase and the consensus phase. In section 4.3, two privacy proofs are done: indistinguishability of an arbitrary change of the initial value on the information gathered by the adversaries and information-theoretic analysis. The first privacy proof is similar to the privacy proof done at [25][36] which proves privacy by analyzing the information output of the adversary. The information that is available to estimate the initial value changes according to the capabilities of the adversary. Main idea is to show that the information the adversaries see can be kept the same when the initial value of the target node changes arbitrarily. The second privacy analysis

starts by defining how much information is leaked to the system at each release of an alpha state. Then, by analyzing the estimator's performance and using the results found in [26], the final privacy of the system is examined. The last chapter gives numerical examples on the convergence and the estimation performance of the proposed approach.

4.2 Methodology

Each node decomposes its state value, say $x_i[0] \in \mathbb{R}$ into two substates $x_i^\alpha[0] \in \mathbb{R}$ and $x_i^\beta[0] = 2x_i[0] - x_i^\alpha[0]$ resulting in an increase in the number of nodes from N to $2N$. $x^\alpha[k]$ is used in the interaction with the other nodes, while $x^\beta[k]$ is used as an internal update. Although $x^\beta[k]$ is never shared, it is used in the evaluation of $x^\alpha[k]$. The randomized gossip update can be written as following.

$$x_i[k+1] = x_i[k] + \frac{1}{2}(x_j[k] - x_i[k]) \tag{4.1}$$

Using the state decomposition approach [25], eq. (4.1) becomes

$$\begin{aligned} x_i^\alpha[k+1] &= x_i^\alpha[k] + \frac{1}{3}(x_j^\alpha[k] - x_i^\alpha[k]) \\ &\quad + \frac{1}{3}(x_i^\beta[k] - x_i^\alpha[k])\,, \\ x_i^\beta[k+1] &= x_i^\beta[k] + \frac{1}{3}(x_i^\alpha[k] - x_i^\beta[k]) \end{aligned} \tag{4.2}$$

subject to $x_i^\alpha[0] + x_i^\beta[0] = 2x_i[0]$.

Forcing the update weights to be between $(0, 1)$ limits the privacy that can be provided. For this reason, two phases are introduced: the initialization phase and the consensus phase. In the initialization phase, the update weights are selected from the set of all real numbers with the condition that the sum of all state variables never changes. Selecting the coupling weights from the set of all real numbers introduce randomness to the system that will provide the privacy of the initial values. The convergence rate does not get affected but the initial errors get bigger. Compared to the randomized gossip, the convergence rate is lower because of the increase in the number of nodes. As the sum of the state variables does not change, the exact consensus can still be achieved.

At consensus phase the update equations are the same as eq. (4.2). As privacy is already established in the initialization phase, the motivation is to let nodes reach the average of their state values in finite time. When v_i goes through the initialization update once with all its neighbors, it proceeds to the consensus phase. Throughout the paper the following assumption holds.

Assumption 1. The graph is connected and there are no channel encryptions in the network.

4.2.1 Initialization Phase

The purpose of this phase is to introduce randomness to the system to preserve the privacy of the initial values. The coupling weights which are fixed to $1/3$ in the eq. (4.2) are selected from the set of all real numbers instead. The

update equations for this phase becomes

$$\begin{aligned} x_i^{\alpha}[k+1] &= x_i^{\alpha}[k] + a_{ij}[k](x_j^{\alpha}[k] - x_i^{\alpha}[k]) + a_{i,\alpha\beta}[k](x_i^{\beta}[k] - x_i^{\alpha}[k])\,, \\ x_i^{\beta}[k+1] &= x_i^{\beta}[k] + a_{i,\alpha\beta}[k](x_i^{\alpha}[k] - x_i^{\beta}[k]) \end{aligned} \tag{4.3}$$

under the constraint that $x_i^{\alpha}[0] + x_i^{\beta}[0] = 2x_i[0]$. The coupling weights, $a_{i,\alpha\beta}[k] \in \mathbb{R}$ and $a_{ij}[k] \in \mathbb{R}$, are drawn independently at each round. In addition to this, they are symmetric at each iteration, i.e. $a_{ij}[k] = a_{ji}[k]$. To achieve this equality, v_i picks $a_{i\to j} \in \mathbb{R}$ randomly and sends it to v_j. The multiplication $a_{ij} = a_{i\to j}a_{j\to i}$ is the shared coupling weight that will be used by v_i and v_j. This choice of design, keeps the sum of all the substates at each iteration the same.

The initialization stage is active unless both nodes have gone through an initialization update and there isn't any neighbor left that the node has not talked to. This choice depends on two reasons. First one is to make sure that the coupling weight that is hidden from the adversary is communicated. If all the neighbors are traversed, it is guaranteed that the coupling weight that the adversary cannot listen to is communicated. The second one is to make the system suitable for the topological way of hiding one of the coupling weight from the adversary. A topological assumption is made in [26][24] which puts the following constraint on the network.

$$N_i \cup v_i \not\subset N_j \cup v_j \tag{4.4}$$

This enables at least one of the coupling weight to be hidden from the adversary however, in the state decomposition the coupling weight needs to be hidden from the adversary only during the initialization phase. Although this assumption would give the desired condition to prove privacy, it can be relaxed to hold only during the initialization phase. For this reason, a theoretical proof is done in the privacy analysis part that assumes one of the coupling weight to be hidden from the adversary. At chapter 5, a cryptographic method that minimizes the number of encryptions is proposed which bases its result on the theoretical privacy proof done at this chapter.

Algorithm 1 Asynchronous State Decomposition Averaging

Let S_i to define the set of neighbors that v_i has selected before.

Let F_i be the set of neighbors of v_i that have finished initialization.

Each node decomposes its state variable into $x_i[0] \in \mathrm{R}$ and $x_i^{\beta}[0] = 2x_i[0] - x_i^{\alpha}[0]$

for k=1,...,K **do**

  Select v_i with probability $p_i = \frac{1}{N}$

  if $S_i \neq N_i$ **then**

    Select $v_j \in N_i \setminus S_i$ with $p_{j|i} = \frac{1}{|N_i \setminus S_i|}$

    Add v_j to S_i

    Select $a_{i\to j} \in R$ and broadcast $x_i^{\alpha}[k], a_{i\to j}$

    Calculate $a_{ij} = a_{i\to j} a_{j\to i}$

    Select $a_{i,\alpha\beta} \in R$

    Update using eq. (4.3)

    Broadcast if $S_i = N_i$

  else if $F_i \neq \varnothing$ **then**

    Select $v_j \in F_i$ with probability $p_{j|i} = \frac{1}{|F_i|}$

    Broadcast $x_i^{\alpha}[k]$

    Update using eq. (4.2)

  else

    Skip the iteration

  end if

end for

4.2.2 Consensus Phase

During the consensus phase, the nodes update their state variables to reach the average of their state variables at the start of the process. During the convergence analysis, the sum is shown to be preserved throughout the algorithm which means that the nodes will reach to the average of their initial variables. The node v_i is selected with equal probability $p_i = 1/N$. If v_i has gone through the initialization update with all its neighbors once, it selects a neighboring node v_j with equal probability $p_{j|i} = \frac{1}{|F_i|}$ where F_i is defined to be the set of neighbors of v_i that have finished the initialization. Given that F_i is not empty, nodes v_i and v_j go through the consensus update defined in eq. (4.2). If there is no neighbor that has finished the initialization, the update is skipped.

As the nodes communicate with each other asynchronously and the selection of nodes are random, the time when the nodes finish the initialization process is not deterministic. The node is said to finish the initialization process when it has gone through initialization update once it all its neighbors. Once it has finished the initialization, the node is ready to go through the consensus update with the neighbors that also have finished the initialization. This is done to reduce the idle time and start the consensus process as fast as possible.

4.2.3 Convergence Analysis

Since the coupling weights are symmetric, the sum of the network does not change at each iteration. It can be shown that for each update during initialization,

$$\frac{1}{2N}\sum_{j=1}^{N}(x_j^{\alpha}[k]+x_j^{\beta}[k]) = \frac{1}{2N}\sum_{j=1}^{N}(x_j^{\alpha}[k+1]+x_j^{\beta}[k+1]).$$

After the initialization, the convergence analysis is based on the paper [20] on constrained consensus and optimization in multi-agent networks and privacy-preserving state decomposition[25]. Four requirements are given in [20] for the exact consensus to be achieved. These properties are shown to hold as follows.

- *Weight Rule:* There exists a scalar η with $0 < \eta < 1$ such that for every iteration after the initialization, all nonzero $a_{ij}[k]$ satisfy $\eta \leq a_{ij}[k] < 1$ and all nonzero $a_{i,\alpha\beta}[k]$ satisfy $\eta \leq a_{i,\alpha\beta}[k] < 1$. In fact, both $a_{ij}[k]$ and $a_{i,\alpha\beta}[k]$ are fixed to $1/3$ which is between $(0, 1)$. The rest of the coupling weights are set to zero.

- *Doubly Stochasticity:* For every update, three out of all the coupling weights are $1/3$ where the rest is zero. Since all the coupling weights are also symmetric, the sum is preserved and $(1, \mathbf{1})$ is an eigenpair of the weight matrix.

- *Connectivity:* The graph G=(V,E) before decomposition consists of a connected network because of the Assumption 1. State decomposition creates a connected graph since each node decomposes itself into two substates which are also connected. Thus, the graph is connected.

- *Bounded Intercommunication Interval:* If Algorithm 1 is followed, the nodes go through the initialization unless they have completed initialization with all its neighbors. As the selection of nodes are with equal probability and they select neighbor nodes that they have not connected, it is expected that all nodes will finish initialization in finite iterations. During the consensus phase, the nodes go through the update with the neighbors that they know who has finished initialization. Since the selection of nodes is at random, in a bounded time B, each node is expected to be contacted at least once.

As four of the requirements hold, all substates will converge to the mean of the $\frac{1}{2N}\sum_{j=1}^{N}(x_j^{\alpha}[k]+x_j^{\beta}[k])$ which is equivalent to $\frac{1}{N}\sum_{j=1} x_j[0]$ due to the initial constraint $x_i^{\alpha}[0]+x_i^{\beta}[0] = 2x_i[0]$.

$$\lim_{k\to\infty} x_i^{\alpha}[k] = \lim_{k\to\infty} x_i^{\beta}[k] = \frac{1}{N}\sum_{j=1} x_j[0]$$

4.3 Privacy

The privacy of the initial value $x_j[0]$ of node v_j will be examined according to two adversaries. First one is a semi-honest adversary who follows the protocol and has knowledge of the communication happening in its range. The second one is the eavesdropper who has the power of intercepting communications happening arbitrarily in the network. The aim of both the adversary is to try to guess the initial value of node v_j with the information that they have.

Definition 4.1. The initial value $x_i[0]$ of the node v_i is said to be private if an adversary does not have any range to estimate the initial value.

With this definition, the privacy claim can be made as follows.

Claim 1. If algorithm 1 is followed, the initial value of node v_j is private to semi-honest node v_i given that there is at least one coupling weight hidden from the adversary during the initialization.

4.3.1 Initial Value Indistinguishability Analysis for Semi-Honest Adversary

The semi-honest adversary is defined to be the node v_i that listens to all the communication of the node v_j, the neighbors of node v_i and has knowledge of the private information that node v_i never shares. This can be represented with the information output of node v_i at the iteration k as,

$$I_i[k] = \{a_{mp}[k]|_{v_p \in N_i, v_m \in N_i, m \neq p}, x_p^{\alpha}[k]|_{v_p \in N_i}, x_i[k], x_i^{\alpha}[k], x_i^{\beta}[k], a_{i,\alpha\beta}[k]\} \tag{4.5}$$

The accumulated information that the node v_i can know about node v_j is $I_i = \cup_{k=0}^{\infty} I_i[k]$. Using this information, claim 1 can be proven.

Proof. The way to prove this statement is to show that the information output of node v_i is indifferent to any different initial value of node v_j. Let the new initial values to be represented with the barred versions of their notations. Thus, the aim is to prove that $I_i = \bar{I}_i$ given that $x_j[0] \neq \bar{x}_j[0]$. Since the exact convergence is to be achieved, it is assumed that the sum of the all nodes initial values have not changed. In other words without loss of generality, the difference $c = x_j[0] - \bar{x}_j[0]$ is assumed to be the difference of $c = \bar{x}_m[0] - x_m[0]$ where node v_m is the neighbour of node v_j that node v_i can't listen to. Let the initial values be the following.

$$\begin{aligned} \bar{x}_m[0] &= x_j[0] + x_m[0] - \bar{x}_j[0] \\ \bar{x}_j^{\alpha}[0] = x_j^{\alpha}[0],\ \bar{x}_j^{\beta}[0] &= 2\bar{x}_j[0] - x_j^{\alpha}[0] \\ \bar{x}_m^{\alpha}[0] = x_m^{\alpha}[0],\ \bar{x}_m^{\beta}[0] &= 2\bar{x}_m[0] - x_m^{\alpha}[0] \\ \bar{x}_q[0] = x_q[0],\ \bar{x}_q^{\alpha}[0] = x_q^{\alpha}[0],\ \bar{x}_q^{\beta}[0] &= x_q^{\beta}[0], \forall v_q \in V \backslash \{v_j, v_m\} \end{aligned} \tag{4.6}$$

As the proposed algorithm includes asynchronous updates, different information outputs are created that are dependent upon the selection of the nodes. Under the condition that all the edges are traversed during the initialization process, it can be said that node v_j will talk to the node v_m that can't be listened by node v_i. Let the coupling weights of node v_j to be the following when it is communicating with the $v_q \in N_j \backslash v_m$.

$$\begin{aligned} \bar{a}_{j,\alpha\beta}[k] &= a_{j,\alpha\beta}[k] \frac{x_j^{\beta}[k] - x_j^{\alpha}[k]}{\bar{x}_j^{\beta}[k] - x_j^{\alpha}[k]} \\ \bar{a}_{jq}[k] &= a_{jq}[k], v_q \in N_j \backslash m \end{aligned} \tag{4.7}$$

The resulting information output that node v_i will observe could be kept the same with the initial values given in eq. (4.6) and the coupling weights given in eq. (4.7). Although the initial value of nodes v_m and v_j have changed, the α states that they reveal will be the same. The effect of the change will accumulate in the β states of these two nodes.

The updates of the nodes other than m and j are trivial and they are the same since all the coupling weights are kept the same. The updates of node v_j will create the same α states in I_i with the given update weights.

$$\begin{aligned}
\bar{x}_j^{\alpha}[k] &= x_j^{\alpha}[k-1] + a_{jq}(x_q^{\alpha}[k-1] - x_j^{\alpha}[k-1]) + \bar{a}_{j,\alpha\beta}[k-1](\bar{x}_j^{\beta}[k-1] - x_j^{\alpha}[k-1]) \\
&= x_j^{\alpha}[k-1] + a_{jq}(x_q^{\alpha}[k-1] - x_j^{\alpha}[k-1]) + a_{j,\alpha\beta}[k-1]\frac{x_j^{\beta}[k-1] - x_j^{\alpha}[k-1]}{\bar{x}_j^{\beta}[k-1] - x_j^{\alpha}[k-1]}(\bar{x}_j^{\beta}[k-1] - x_j^{\alpha}[k-1]) \\
&= x_j^{\alpha}[k-1] + a_{jq}(x_q^{\alpha}[k-1] - x_j^{\alpha}[k-1]) + a_{j,\alpha\beta}[k-1](x_j^{\beta}[k-1] - x_j^{\alpha}[k-1]) \\
&= x_j^{\alpha}[k]
\end{aligned}$$

The β states that are never released will be different with the new initial values. This difference will be compensated when the node v_j is talking to the neighbor node v_m that the semi-honest adversary node v_i can't listen to.

$$\begin{aligned}
\bar{x}_j^{\beta}[k] &= \bar{x}_j^{\beta}[k-1] + \bar{a}_{j,\alpha\beta}[k-1](x_j^{\alpha}[k-1] - \bar{x}_j^{\beta}[k-1]) \\
&= \bar{x}_j^{\beta}[k-1] + a_{j,\alpha\beta}[k-1]\frac{x_j^{\beta}[k-1] - x_j^{\alpha}[k-1]}{\bar{x}_j^{\beta}[k-1] - x_j^{\alpha}[k-1]}(x_j^{\alpha}[k-1] - \bar{x}_j^{\beta}[k-1]) \\
&= \bar{x}_j^{\beta}[k-1] + a_{j,\alpha\beta}[k-1](x_j^{\alpha}[k-1] - x_j^{\beta}[k-1])
\end{aligned}$$

Let the coupling weights of node v_j to be the following when it is communicating with the node v_m that node v_i can't listen to.

$$\begin{aligned}
\bar{a}_{j,\alpha\beta}[k] &= \frac{x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{j,\alpha\beta}[k](x_j^{\alpha}[k] - x_j^{\beta}[k])}{(x_j^{\alpha}[k] - \bar{x}_j^{\beta}[k])} \\
\bar{a}_{jm}[k] &= \frac{x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{jm}[k](x_m^{\alpha}[k] - x_j^{\alpha}[k])}{(x_m^{\alpha}[k] - x_j^{\alpha}[k])}
\end{aligned} \tag{4.8}$$

With the given weights and the initial values, $\bar{I}_i$ that the semi-honest adversary will observe until the $k+1$th iteration will be the same as I_i regardless of $x_j[0]$ or $\bar{x}_j[0]$.

$$\begin{aligned}
\bar{x}_j^{\alpha}[k+1] &= x_j^{\alpha}[k] + \bar{a}_{jm}(x_m^{\alpha}[k] - x_j^{\alpha}[k]) + \bar{a}_{j,\alpha\beta}[k](\bar{x}_j^{\beta}[k] - x_j^{\alpha}[k]) \\
&= x_j^{\alpha}[k] + \frac{x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{jm}[k](x_m^{\alpha}[k] - x_j^{\alpha}[k])}{x_m^{\alpha}[k] - x_j^{\alpha}[k]}(x_m^{\alpha}[k] - x_j^{\alpha}[k]) \\
&\quad + \frac{x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{j,\alpha\beta}[k](x_j^{\alpha}[k] - x_j^{\beta}[k])}{(x_j^{\alpha}[k] - \bar{x}_j^{\beta}[k])}(\bar{x}_j^{\beta}[k] - x_j^{\alpha}[k]) \\
&= x_j^{\alpha}[k] + x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{jm}[k](x_m^{\alpha}[k] - x_j^{\alpha}[k]) + x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{j,\alpha\beta}[k](x_j^{\alpha}[k] - x_j^{\beta}[k]) \\
&= x_j^{\alpha}[k+1]
\end{aligned}$$

In addition to this, all the sub-states of the nodes that the adversary can listen to will be also the same.

$$\begin{aligned}
\bar{x}_j^{\beta}[k+1] &= \bar{x}_j^{\beta}[k] + \bar{a}_{j,\alpha\beta}[k](x_j^{\alpha}[k] - \bar{x}_j^{\beta}[k]) \\
&= \bar{x}_j^{\beta}[k] + \frac{x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{j,\alpha\beta}[k](x_j^{\alpha}[k] - x_j^{\beta}[k])}{(x_j^{\alpha}[k] - \bar{x}_j^{\beta}[k])}(x_j^{\alpha}[k] - \bar{x}_j^{\beta}[k]) \\
&= \bar{x}_j^{\beta}[k] + x_j^{\beta}[k] - \bar{x}_j^{\beta}[k] + a_{j,\alpha\beta}[k](x_j^{\alpha}[k] - x_j^{\beta}[k]) \\
&= x_j^{\beta}[k+1]
\end{aligned}$$

It is necessary at this point that the symmetric update weight a_{jm} is the same for both j and m. This demands the following condition to be true.

$$\begin{aligned} a_{jm}[k] &= a_{mj}[k] \\ \frac{x_j^\beta[k] - \bar{x}_j^\beta[k] + a_{jm}[k](x_m^\alpha[k] - x_j^\alpha[k])}{(x_m^\alpha[k] - x_j^\alpha[k])} &= \frac{x_m^\beta[k] - \bar{x}_m^\beta[k] + a_{mj}[k](x_j^\alpha[k] - x_m^\alpha[k])}{(x_j^\alpha[k] - x_m^\alpha[k])} \\ x_j^\beta[k] - \bar{x}_j^\beta[k] + a_{jm}[k](x_m^\alpha[k] - x_j^\alpha[k]) &= \bar{x}_m^\beta[k] - x_m^\beta[k] - a_{mj}[k](x_j^\alpha[k] - x_m^\alpha[k]) \\ x_j^\beta[k] - \bar{x}_j^\beta[k] &= \bar{x}_m^\beta[k] - x_m^\beta[k] \end{aligned} \tag{4.9}$$

However, the initialization of the nodes depends on the order of the selection of the nodes. This requires eq. (4.9) to hold regardless of how the nodes are initiated. An example case can be the following sequence of initialization: v_j talks to v_q which can be eavesdropped by v_i first. Then, v_j talks to v_m which v_i can't eavesdrop. Firstly, the resulting state will be the following using the update weights eq. (4.7).

$$\begin{aligned} \bar{x}_j^\alpha[1] &= x_j^\alpha[1] \\ \bar{x}_j^\beta[1] &= \bar{x}_j^\beta[0] + \bar{a}_{j,\alpha\beta}[0](x_j^\alpha[0] - \bar{x}_j^\beta[0]) \\ &= \bar{x}_j^\beta[0] + a_{j,\alpha\beta}[0]\frac{x_j^\beta[0] - x_j^\alpha[0]}{\bar{x}_j^\beta[0] - x_j^\alpha[0]}(x_j^\alpha[0] - \bar{x}_j^\beta[0]) \\ &= \bar{x}_j^\beta[0] + a_{j,\alpha\beta}[0](x_j^\alpha[0] - x_j^\beta[0]) \end{aligned} \tag{4.10}$$

Secondly, v_j will be communicating with v_m. The requirement for the symmetric update weight a_{jm} to be the same, the following condition needs to hold.

$$x_j^\beta[1] - \bar{x}_j^\beta[1] = \bar{x}_m^\beta[0] - x_m^\beta[0] \tag{4.11}$$

This is the same result that was obtained in eq. (4.9). When the nodes are not talking to each other, they keep their states for the next iteration. As v_j has gone through the update once and v_m has not, the condition in eq. (4.9) transfer to the equation above. Using the eq. (4.10) and plugging it into eq. (4.11),

$$\begin{aligned} x_j^\beta[0] + a_{j,\alpha\beta}[0](x_j^\alpha[0] - x_j^\beta[0]) - (\bar{x}_j^\beta[0] + a_{j,\alpha\beta}[0](x_j^\alpha[0] - x_j^\beta[0])) &= \bar{x}_m^\beta[0] - x_m^\beta[0] \\ x_j^\beta[0] - \bar{x}_j^\beta[0] &= \bar{x}_m^\beta[0] - x_m^\beta[0] \end{aligned} \tag{4.12}$$

It can be seen that the first equation at eq. (4.6) proves that the result in eq. (4.12) will make a_{jm} symmetric. Irrespective of the way the nodes are selected, the result would be the same. Using the $a_{j,\alpha\beta}[k]$ at eq. (4.8), the difference $x_j^\beta[k] - \bar{x}_j^\beta[k]$ can be kept the same. The $k+1$th iteration where v_j talks with v_q will be,

$$\begin{aligned} \bar{x}_j^\beta[k+1] &= \bar{x}_j^\beta[k] + \bar{a}_{j,\alpha\beta}[k](x_j^\alpha[k] - \bar{x}_j^\beta[k]) \\ &= \bar{x}_j^\beta[k] + a_{j,\alpha\beta}[k]\frac{x_j^\beta[k] - x_j^\alpha[k]}{\bar{x}_j^\beta[k] - x_j^\alpha[k]}(x_j^\alpha[k] - \bar{x}_j^\beta[k]) \\ &= \bar{x}_j^\beta[k] + a_{j,\alpha\beta}[k](x_j^\alpha[k] - x_j^\beta[k]) \end{aligned}$$

The difference $x_j^\beta[k+1] - \bar{x}_j^\beta[k+1]$ will be the same as $x_j^\beta[k] - \bar{x}_j^\beta[k]$.

$$\begin{aligned} x_j^\beta[k+1] - \bar{x}_j^\beta[k+1] &= x_j^\beta[k] + a_{j,\alpha\beta}[k](x_j^\alpha[k] - x_j^\beta[k]) - \bar{x}_j^\beta[k] - a_{j,\alpha\beta}[k](x_j^\alpha[k] - x_j^\beta[k]) \\ &= x_j^\beta[k] - \bar{x}_j^\beta[k] \end{aligned}$$

The same can be told for v_m. As the difference stays the same, the first condition of eq. (4.6) would suffice to make the $a_{jm}[k]$ symmetric when they are communicating with each other regardless of the earlier updates.

$$\begin{aligned} x_j^\beta[k+1] - \bar{x}_j^\beta[k+1] &= \bar{x}_j^\beta[k] - x_j^\beta[k] \\ &= \bar{x}_j^\beta[0] - x_j^\beta[0] = x_m^\beta[0] - \bar{x}_m^\beta[0] \end{aligned}$$

As the α and β states of all the nodes in the network can be kept the same after the initialization phase, the consensus phase will create the same $I_i[k]$. This proves that if there is at least one neighbor of v_j that semi-honest adversary v_i can't listen to, the adversary can't estimate the initial value of v_j with any range as the information outputs I_i and $\bar{I}_i$ will be the same. □

4.3.2 Information-theoretic Privacy Analysis for the Semi-Honest Adversary

In information theory, mutual information has been widely used as a way to describe the dependency between two random variables. Higher mutual information means that there is a great correlation between the two random variables whereas lower mutual information means there is less dependency between the two random variables. The estimation of mutual information via the realizations of random variables has been a topic of interest and methods such as histogram binning or non-parametric estimation using k-nearest neighbor statistics have been proposed[49]. It has been found that if there are enough samples, the non-parametric estimates of mutual information give low bias results. In addition to this, the histogram binning is very sensitive to the bin sizes that are used. For these reasons, non-parametric estimation is used to estimate the mutual information. More information on this method can be found in section 3.5.

Contrary to indistinguishability analysis of section 4.3.1, information theory analyzes privacy through the correlation of random variables. The main motivation is to show that there is no correlation between the information output of the adversary and the initial value of the observed node. Let the semi-honest adversary to be denoted as v_i and the observed node to be denoted as v_j. The information output of v_i is the same eq. (4.5). Given that the adversary can listen to v_j and all its neighbors, it can estimate the initial value exactly as following. Let $s[k]$ denote the sum $\sum_{k=1}^{T} a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ that can be obtained by

$$s[k+1] = s[k] + x_j^\alpha[k+1] - a_{jp}[k](x_p^\alpha[k] - x_j^\alpha[k]) \tag{4.13}$$

where $s[0] = x_j^\alpha[0]$, T is the iteration that v_j has finished the initialization phase and $a_{jp}[k]$ is the coupling weight between v_j and $v_p \in N_j$. The initial value can be found by the following relation

$$\hat{x}_j[0] = \frac{1}{2}(s[T] + x_j^\beta[T]) \tag{4.14}$$

as the first consensus update after the initialization discloses $x_j^\beta[T]$ due to the fixed coupling weights. This disclosure can be shown by observing the first alpha state variable consensus update.

$$\begin{aligned} x_j^\alpha[T+1] &= x_j^\alpha[T] + \frac{1}{3}(x_p^\alpha[T] - x_j^\alpha[T]) + \frac{1}{3}(x_j^\beta[T] - x_j^\alpha[T]) \\ x_j^\beta[T] &= 3x_j^\alpha[T+1] - x_j^\alpha[T] - x_p^\alpha[T] \end{aligned}$$

The assumption that there is at least one coupling weight hidden from the adversary, enables one of the $a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ to be blinded by $a_{jp}[k](x_p^\alpha[k] - x_j^\alpha[k]), v_p \in N_j$, disrupting the deterministic property of the estimator update

given in eq. (4.14). This is a necessary condition to prove the privacy. It is important to notice that if $(x_j^\beta[k] - x_j^\alpha[k])$ is 0, the adversary will know exactly what $a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ is. In this case, the adversary will know the initial value exactly and there will be no privacy. However, as α states variables are selected from the set of all real numbers, possibility of this happening is negligibly small.

To satisfy the privacy given in definition 4.1, two cases need to be investigated. First one is the information leakage when node v_j is communicating a neighboring node that can be listened by the adversary and the second one is when v_j is communicating the neighboring node that cannot be listened by the adversary. There will be at most $N_j - 1$ updates within the neighborhood of the adversary and there will be at least one update outside the neighborhood of the adversary. Due to the asynchronous nature of the algorithm, the order at which these updates will be done is not fixed. However, if algorithm 1 is followed, it is guaranteed that v_j will go through the initialization update with all its neighbors including the one outside the neighborhood of the adversary. For the first case, the aim is to hide the initial value from the adversary while for the second case, the aim is to hide $a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ to break the deterministic nature of the estimator eq. (4.14).

The first case is when v_j is contacting the nodes that can be listened by the adversary. The goal is to hide the initial value from the adversary. As $x_j^\alpha[0]$ is released and known, $x_i^\beta[0]$ is the only information required to get to $x_j[0]$ through the relation $2x_j[0] = x_j^\alpha[0] + x_j^\beta[0]$. It can be said that the secrecy of $x_j[0]$ is equivalent to the secrecy of $x_j^\beta[0]$. In addition to $x_j^\alpha[0]$, the variables $x_p^\alpha[0]$ and $a_{jp}[0]$ are also known. For this case, the information leakage can be defined as following.

$$I(x_j^\alpha[1]; x_j^\beta[0] | x_j^\alpha[0], a_{jp}[0], x_p^\alpha[0])$$

Using the eq. (4.3) expression becomes,

$$I(x_j^\alpha[1]; x_j^\beta[0] | x_j^\alpha[0], a_{jp}[0], x_p^\alpha[0]) = I(a_{j,\alpha\beta}[0](x_j^\beta[0] - x_j^\alpha[0]); x_j^\beta[0] | x_j^\alpha[0])$$

The internal coupling weight $a_{j,\alpha\beta}[0]$ acts like a multiplicative noise whereas, the $x_j^\alpha[0]$ acts as an additive noise. A numerical analysis is done to investigate the effect of various input distributions and their variances on $I(x_j^\alpha[1]; x_j^\beta[0] | x_j^\alpha[0])$. This is an analysis of the mutual information regarding the first case where

The estimation of conditional mutual information is based on the papers [49], [50] and the non-parametric entropy estimator toolbox [48]. Assuming that the input distribution, $x_j^\alpha[0]$ distribution and the coupling weights are sampled from a continuous uniform distribution, the effect of increasing the variance of the coupling weight $a_{j,\alpha\beta}[0]$ and $x_j^\alpha[0]$ can be seen in fig. 4.1. The initial values are selected from a unit variance uniform distribution, $U[-\sqrt{3}, \sqrt{3}]$, and the estimates have been done using $N = 10^5$ realizations. The non-monotonic nature of the estimates is due to the bias errors introduced by estimating the mutual information. However, it can be seen that as the variance increases, the conditional mutual information gets closer to zero. Although $x_j^\alpha[0]$ is a constant, it still introduces randomness to the multiplication $a_{j,\alpha\beta}[0](x_j^\beta[0] - x_j^\alpha[0])$. It is necessary to include the effect of $x_i^\alpha[0]$ to the conditional mutual information calculation as it is multiplied by $a_{j,\alpha\beta}[0]$ and added to the weighted secret $x_j^\beta[0]$.

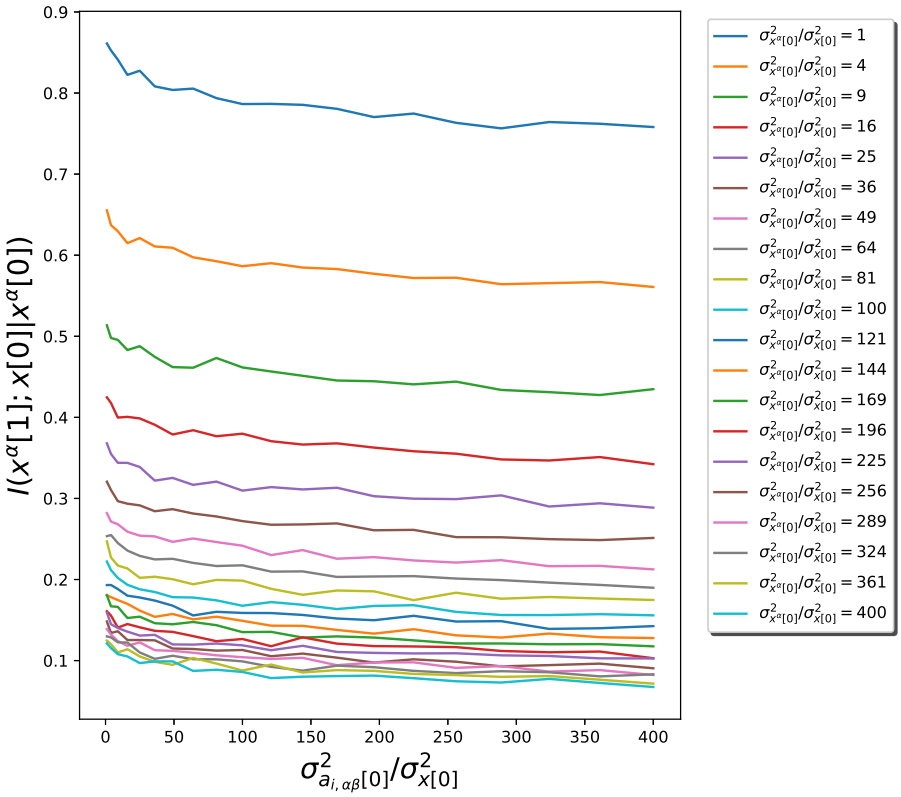

Figure 4.1: Conditional Mutual Information $I(x_i^\alpha[1]; x_i^\beta[0]|x_i^\alpha[0])$ for increasing $a_{i,\alpha\beta}[0]$ and $x_i^\alpha[0]$ variance where all random variables are sampled from uniform distributions

	Input Distribution		
	Uniform	Laplacian	Gaussian
$a_{i,\alpha\beta}[0]$	Laplacian	Laplacian	Laplacian
$x_i^\alpha[0]$	Laplacian	Uniform	-

Table 4.1: The distributions that minimize the conditional mutual information for various input distributions

The effect of choosing different distributions for $x_j^\alpha[0]$ and coupling weights have been examined to find the one that minimizes the $I(x_j^\alpha[1]; x_j^\beta[0]|x_j^\alpha[0])$. For this reason three possible input value distributions are selected namely laplacian, gaussian and uniform distributions. $x_j^\alpha[0]$ and $x_j[0]$ is selected to be unit variance and the variance ratio $\sigma^2_{a_{j,\alpha\beta}}/\sigma^2_x$ is increased to understand which distribution combination results in the least conditional mutual information.

For the uniformly distributed initial values, fig. 4.2 shows that laplacian distributed $a_{j,\alpha\beta}[0]$ and $x_j^\alpha[0]$ gives the

least conditional mutual information. For gaussian distributed initial values, fig. 4.3 shows that laplacian distributed $a_{j,\alpha\beta}[0]$ regardless of the distribution of $x_j^\alpha[0]$ gives the least conditional mutual information. For laplacian distributed initial values fig. 4.4 shows that laplacian distributed $a_{j,\alpha\beta}[0]$ and uniformly distributed $x_j^\alpha[0]$ give the best results. Overall it has been observed that the laplacian $a_{j,\alpha\beta}[0]$ results in the best blinding of $x_j[0]$. The summary of the best results can be seen in table 4.1. With a priory knowledge of the input distribution, the coupling weights and alpha states can be selected from the distributions that would minimize the mutual information.

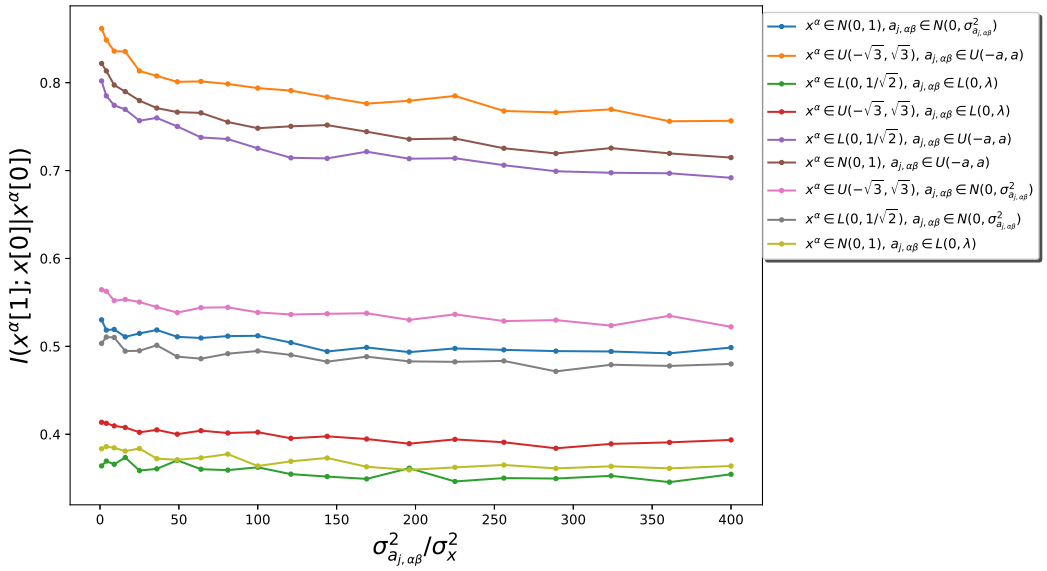

Figure 4.2: $I(x_j^\alpha[1]; x_j^\beta[0]|x_j^\alpha[0])$ for different distributions given a unit variance uniformly distributed $x_j[0]$

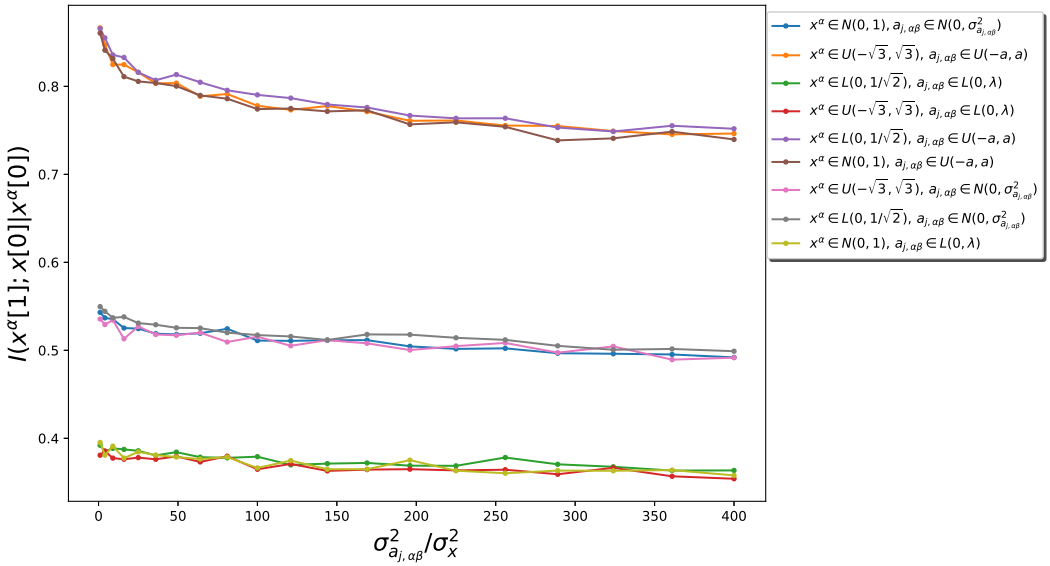

Figure 4.3: $I(x_j^\alpha[1]; x_j^\beta[0]|x_j^\alpha[0])$ for different distributions given a unit variance gaussian distributed $x_j[0]$

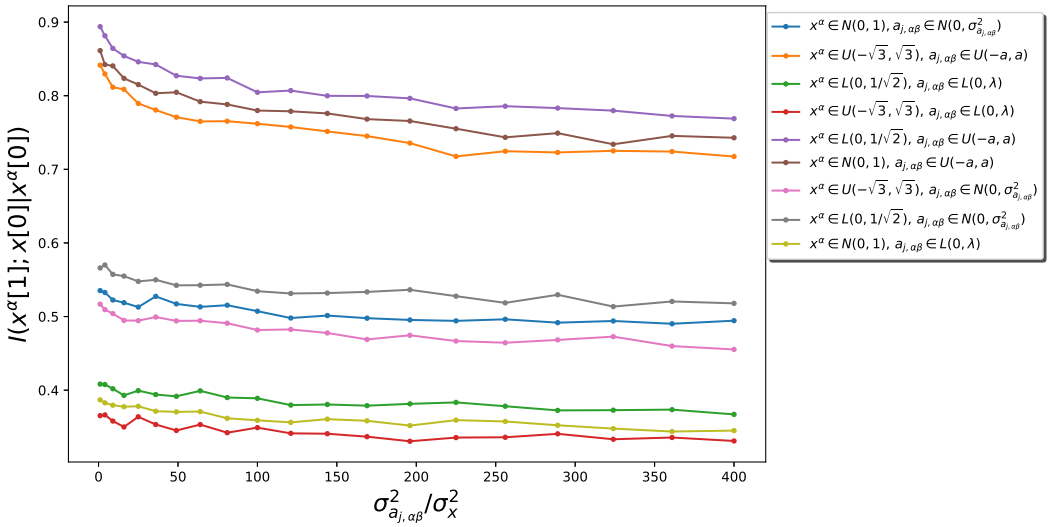

Figure 4.4: $I(x_j^\alpha[1]; x_j^\beta[0]|x_j^\alpha[0])$ for different distributions given a unit variance laplacian distributed $x_j[0]$

An additional test has been made which compares the effect of choosing the sample size N. The $x_j^\alpha[0]$ and $x_j[0]$ is selected to be unit variance and the variance ratio $\sigma^2_{a_{j,\alpha\beta}}/\sigma^2_x$ is again increased to understand the effect of it in the estimation of mutual information. Two plots are presented in **??** where the left one represent the results for $N = 10^4$ and right represent the results for $N = 10^5$. Each plot contains the expression of $I(x_j^\alpha[1]; x_j^\beta[0]|x_j^\alpha[0])$ for three

distributions that $x_j^\alpha[0]$, $x_j[0]$ and $\sigma^2_{a_{j,\alpha\beta}}/\sigma_x^2$ share. The first one includes all three distributions to be sampled from laplacian, second to be sampled from gaussian and third from uniform distributions. The results suggest that the sample size effects how fast the mutual information decreases.

Numerical experiments suggest that there is a decrease in the mutual information with an increasing coupling weight variance. A theoretical analysis will be done to prove this case and the privacy claim given in claim 1.

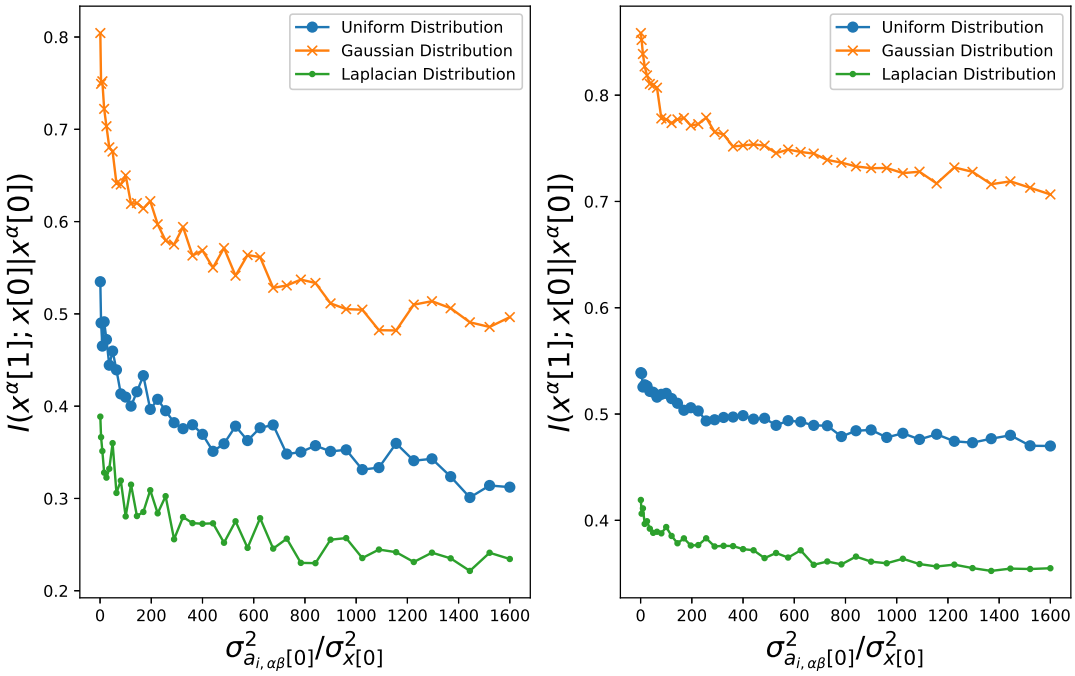

Figure 4.5: Plot of $I(x_j^\alpha[1];x_j^\beta[0]|x_j^\alpha[0])$ for increasing $\sigma^2_{a_{j,\alpha\beta}}/\sigma_x^2$ where all three distributions are sampled from laplacian, gaussian and uniform. The left represent the mutual information for $N = 10^4$ samples and right represent the mutual information for $N = 10^5$.

Proof of claim 1. For the privacy to be established, $x_j^\beta[k]$ should be obfuscated at each iteration and $a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ should be blinded by $a_{ji}[k](x_i^\alpha[k] - x_j^\alpha[k])$. First, it will be shown that for a fixed bounded variance $x_j^\alpha[0]$, the conditional mutual information $I(x_j^\alpha[k+1]);x_j^\beta[k]|x_j^\alpha[k])$ goes to zero as the variance of $a_{j,\alpha\beta}[k]$ goes to infinity. Let $x_j^\alpha[0]$ be a continuous random variable with $\sigma^2_{x_j^\alpha[0]} < \infty$. Define $\gamma = \frac{1}{\sigma^2_{a_{j,\alpha\beta}[k]}}$ and define $\bar{W}_{\alpha\beta} = \gamma a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ with unit variance. The conditional mutual information can be written as the following. The iteration $[k]$ is omitted for clearer notations. Whenever the α and β states, the coupling weights $a_{ji}[k]$ and $a_{i,\alpha\beta}[k]$ are shown without the iteration number, it means the kth iteration. If it is any other iteration than kth iteration, it is explicitly written.

$$I(x_j^\alpha[k+1];x_j^\beta|x_j^\alpha, a_{ji}, x_j^\alpha) = I(W_{\alpha\beta};x_j^\beta|x_j^\alpha)$$

The mutual information is invariant to scaling.

$$I(\gamma W_{\alpha\beta}; \gamma x_j^\beta | \gamma x_j^\alpha) = I(\bar{W}_{\alpha\beta}; \gamma x_j^\beta | \gamma x_j^\alpha)$$

As the variance of $a_{j,\alpha\beta}[k]$ goes to infinity, the conditional mutual information will go to zero.

$$\begin{aligned}\lim_{\sigma^2_{a_{j,\alpha\beta}} \to \infty} I(\bar{W}_{\alpha\beta}; \gamma x_j^\beta | \gamma x_j^\alpha) &= \lim_{\gamma \to 0} I(\bar{W}_{\alpha\beta}; \gamma x_j^\beta | \gamma x_j^\alpha) \\ &= I(\bar{W}_{\alpha\beta}; 0) = 0\end{aligned}$$

Second case is the update when the node that cannot be listened to is contacted. The requirement that there is a one hidden coupling weight, lets $a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ to be obfuscated by $a_{ji}[k](x_i^\alpha[k] - x_j^\alpha[k])$. For this case, $a_{ji}[k]$ and $x_j^\alpha[k]$ is unknown. This prevents the corrupted nodes from learning one of the $a_{j,\alpha\beta}[k](x_j^\beta[k] - x_j^\alpha[k])$ which would have been used to estimate $x_j^\beta[0]$. Define $\beta = \frac{1}{\sigma^2_{a_{ji}[k]}}$ and $\bar{W}_{ij} = \beta a_{ji}[k](x_j^\beta[k] - x_j^\alpha[k])$. Mutual information is invariant to scaling.

$$\begin{aligned}I(x_j^\alpha[k+1]; W_{\alpha\beta} | x_j^\alpha) &= I(\gamma\beta x_j^\alpha[k+1]; \gamma\beta W_{\alpha\beta} | \gamma\beta x_j^\alpha) \\ &= I(\beta\bar{W}_{\alpha\beta} + \gamma\bar{W}_{ij}; \beta\bar{W}_{\alpha\beta} | \gamma\beta x_j^\alpha)\end{aligned}$$

When the variances of both coupling weights go to infinity, the conditional mutual information will go to zero.

$$\begin{aligned}\lim_{\substack{\sigma^2_{a_{ji}} \to \infty \\ \sigma^2_{a_{j,\alpha\beta}} \to \infty}} I(\beta\bar{W}_{\alpha\beta} + \gamma\bar{W}_{ij}; \beta\bar{W}_{\alpha\beta} | \gamma\beta x_j^\alpha) &= \lim_{\substack{\gamma \to \infty \\ \beta \to \infty}} I(\beta\bar{W}_{\alpha\beta} + \gamma\bar{W}_{ij}; \beta\bar{W}_{\alpha\beta} | \gamma\beta x_j^\alpha) \\ &= I(0; 0) = 0\end{aligned}$$

Let T be the iteration at which the initialization has ended. $\mathbf{x}^\alpha[T]$ represents the vector of alpha values obtained starting from $\mathbf{x}[0]$. During the consensus phase let W^k denote the information obtained at each iteration to deduce $\mathbf{x}[0]$ with $k = \{1, 2, ...K\}$ where K is the total iteration number. The total mutual information thus be represented as $I(\mathbf{x}[0]; W^k)$. Fixing the update weights enables to find a function $F^k(\mathbf{x}^\alpha[T]) = W^k$ that will take the $\mathbf{x}^\alpha[T]$ as input and will create the output W^k. The random variables will create a Markov chain $\mathbf{x}[0] \to \mathbf{x}^\alpha[T] \to W^k$ for $k = \{1, 2, ...K\}$. Thus, data processing inequality suggests that,

$$I(\mathbf{x}[0]; \mathbf{x}^\alpha[T]) \geq I(\mathbf{x}[0]; W^k) \text{ for } k = 1, ..., K$$

$I(\mathbf{x}[0]; \mathbf{x}^\alpha[T])$ is shown to be going to zero earlier. Any clever manipulation of data can't increase mutual information. Thus, given that there is at least one neighbor of node v_j whose shared coupling weight is hidden from the adversary, it can't estimate the initial value of node v_j with any guaranteed accuracy.

4.3.3 Eavesdropper

The privacy proof against a semi-honest adversary shows that if one of the coupling weight is hidden between the adversary and its neighbor, there is no range to estimate the initial values. The eavesdropper is a stronger adversary as it can also tap messages that are sent arbitrarily anywhere in the network. Although the information that the eavesdropper has is greater than the semi-honest adversary, the provided privacy is the same under the constraint that one of the coupling weight is unknown. The privacy is provided with this constraint and no more additional information

can be gained if eavesdropper knows any other shared data. Let the information output of the eavesdropper who is capable of tapping every communication channel be the following.

$$I_e[k] = \{a_{ji}[k]|_{v_j \in N, v_i \in N, j \neq i}, x_j^\alpha[k]|_{v_j \in N}\} \tag{4.15}$$

If an eavesdropper intercepts all the communication happening between node v_j and its neighbors, it can estimate the initial value of the node v_j. Let the variable s be defined as the following.

$$s[k+1] = s[k] + x_j^\alpha[k+1] - a_{ij}(x_i^\alpha[k] - x_j^\alpha[k])$$

With $s[0]$ defined to be $x_j^\alpha[0]$, the estimator for the initial value of node v_j, $x_j[0]$, will be the following.

$$\hat{x}_j[0] = \frac{1}{2}(s[k] + x_j^\alpha[k])$$

The system isn't private against an eavesdropper who taps all the communications. Given that there is one coupling weight hidden from it, the same privacy analysis which is done in the section 4.3 can be done also for this case hence, the algorithm will preserve privacy.

4.4 Numerical Examples

The proposed algorithm reaches the exact average consensus of the network while preventing estimation of the initial values with any range. The performance of the asynchronous state decomposition is examined using 5 nodes connected as a circular graph which can be seen in fig. 4.6. This graph is selected such that a comparison can be done with the noise-obfuscation methods which require the topological assumption given at eq. (4.4) to be made on the graph. Circular graph satisfies this property for each node. Geometric graphs can also be used as they have found to represent the wireless sensor networks better[51]. The distance $\sqrt{2 \log n / n}$ that guarantees connectivity with high probability however, creates topological connections which fails the condition given eq. (4.4). For these reasons, the analysis is done on the circular graph. First, the system is shown to converge to the exact average of the initial values. The

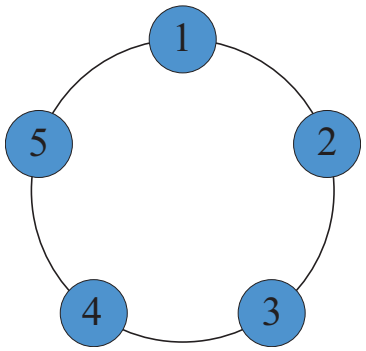

Figure 4.6: Network topology

adjacency matrix of the proposed network is the following.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The initial values of the nodes are selected from $[1, 20]$. The coupling weights and the initial alpha state variables are selected from $[-5, 5]$ during the initialization. It can be seen in fig. 4.7 that the exact consensus can be achieved. The plot on the left shows the convergence of the alpha state variables whereas, the plot on the right show the mean squared error, plotted on a logarithmic scale. Next, the convergence rate of the proposed algorithm compared with the standard

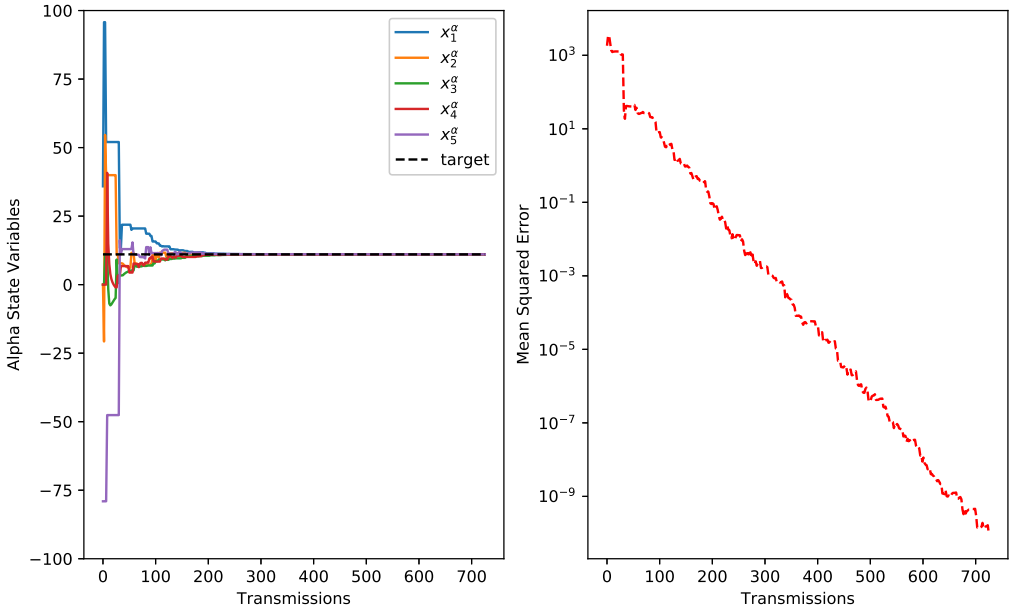

Figure 4.7: The convergence plot for the 5 node circular graph

randomized gossip is examined. It can be seen in fig. 4.8, the proposed approach has a lower rate of convergence. This is due to the increase in the number of nodes from N to $2N$ because of the state decomposition. Finally, an experiment regarding the performance of the estimator is done. The estimator follows the update given in eq. (4.14). One of the coupling weight $a_j p[k]$ is assumed to be unknown to the semi-honest adversary. This prevents the adversary from finding one of the $a_{j,\alpha\beta}[k](x_j^{\beta}[k] - x_j^{\alpha}[k])$ using eq. (4.13). The adversary guesses $a_{j,\alpha\beta}[k](x_j^{\beta}[k] - x_j^{\alpha}[k])$ to be 0 for this case as it has been observed that it has the highest probability to be around 0 if uniform random variables are used for the initialization. The performance of the semi-honest adversary is observed by examining the variance of its estimations. The network topology is the same circular graph given in fig. 4.6. The initial values are selected randomly

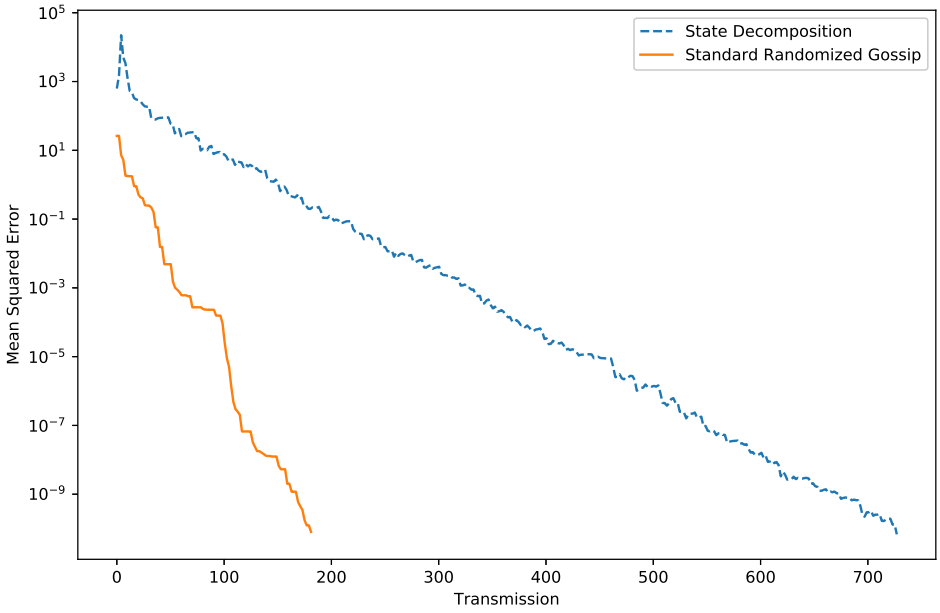

Figure 4.8: Convergence rate plot for the proposed approach and the standard randomized gossip.

from a uniform distribution, $U(0,1)$. The coupling weights, $a_{jp}[k]$ and $a_{j\alpha\beta}[k]$, are selected randomly from a uniform distribution $U(0,\sigma_r^2)$ where σ_r^2 is selected to be $(1,4,9,...,400)$ consecutively. The semi-honest adversary is assumed to be v_2 that observes and guesses the initial value of v_1. There has been 10^4 different initializations for each σ_r^2. For each initialization, the discrepancy between the final estimate and the actual initial value of v_1 is recorded. The variance of these discrepancies is calculated to see if an increase in the coupling weight variance indeed makes the variance of the estimator increase. To estimate the variance from the 10^4 iid initializations, the following unbiased estimator with bessel correction[52] is used.

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{4.16}$$

If the difference between the estimate and the actual initial value of v_1 is plugged into section 6.4, the following equation is obtained.

$$\frac{\sum_{i=1}^{10^4}(x_1^i[0] - \hat{x}_1^i[0])^2}{10^4 - 1} - \frac{(\sum_{i=1}^{10^4}(x_1^i[0] - \hat{x}_1^i[0]))^2}{(10^4-1)10^4} \tag{4.17}$$

where the superscript i at $x_1^i[0]$ and $\hat{x}_1^i[0]$ represent the ith independent initialization and ith independent estimate of the initial value respectively. The result can be seen in fig. 4.9. A linear increase in the estimator's variance is observed as the variances of both coupling weight increases. If the slope is observed, it can be seen that there is an approximate $2:1$ ratio between the estimator variance and σ_r^2. The reason is due to the following properties[53] about

the arithmetic operations of two independent random variables with mean 0:

$$\begin{aligned} Var(XY) &= Var(X)Var(Y)\,, \\ Var(X+Y) &= Var(X)+Var(Y)\,, \\ Var(c*X) &= c^2 Var(X)\,. \end{aligned} \tag{4.18}$$

The α state variables are sampled from a uniform distribution with variance 1 and $a_{j\alpha\beta}[k]$ is sampled with an increasing variance σ_r^2. The sum $(x_j^\beta[0]-x_j^\alpha[0])$ can also be written as $2*(x_j[0]-x_j^\alpha[0])$ using the initial state variable constraint. As there is $1/2$ in the eq. (4.14), there will be no increase due to the multiplication with 2. The sum of $(x_j[0]-x_j^\alpha[0])$ will have a variance of 2. This is multiplied by $a_{j\alpha\beta}[k]$ which makes the variance $2*\sigma_r^2$ where σ_r^2 represent the variance of $a_{j\alpha\beta}[k]$. For this reason, $2:1$ slope is visible in fig. 4.9.

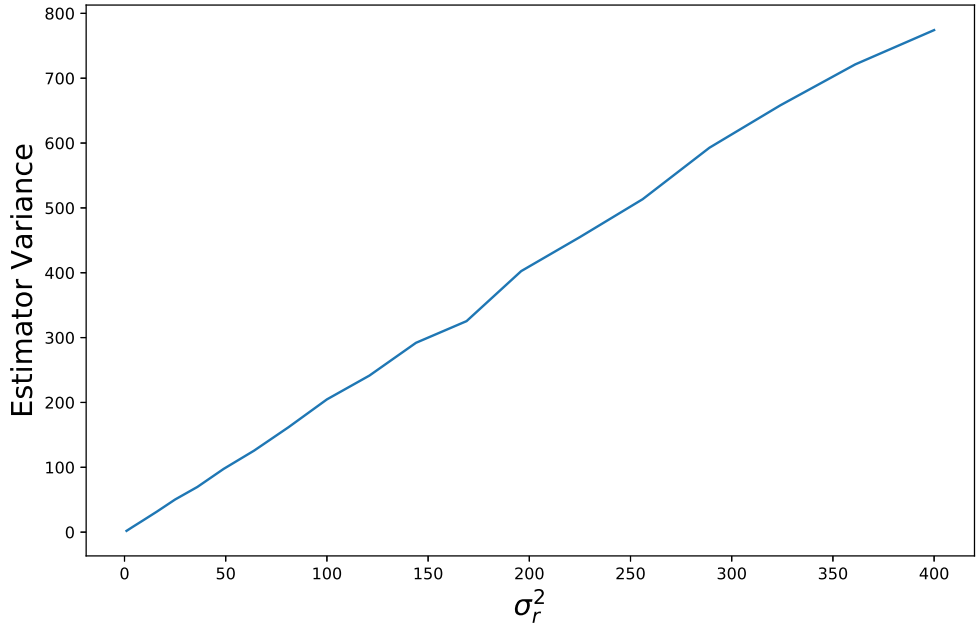

Figure 4.9: The variance of the difference between the estimate and the actual $x_1[0]$ for an increasing σ_r^2 representing the variance of the coupling weights.

The performance of the estimator is examined when the variance of $x_i^\alpha[0]$ is increased as well as the coupling weights. The increase of the variance of the alpha state is the same as the coupling weights which is $(1, 4, 9, ..., 400)$. The variance is calculated with eq. (4.17) under the assumption that adversary guesses 0 again for one of the $a_{j,\alpha\beta}[k](x_j^\beta[k]-x_j^\alpha[k])$ because under uniform distribution assumption, around 0 is where this quantity exists with highest probability. The results can be seen in the fig. 4.10. The increase in the variance is exponential since $x_i^\alpha[0]$ also has an increasing variance. The variance of the coupling weight is multiplied by the initial value which has a linear order in terms of σ_r^2. On the other hand, the multiplication of $a_{j\alpha\beta}[k]x_j^\alpha[k]$ has an exponential order in terms of σ_r^2 due to the multiplicative

property given in eq. (4.18). The sum of $a_{j\alpha\beta}[0]x_j^{\alpha}[0]$ and $a_{j\alpha\beta}[0]x_j[0]$ terms will result in an exponential order in terms of σ_r^2 due to the former term. Thus, the estimator's variance is dominated by $(\sigma_r^2)^2$ which is seen in the fig. 4.10.

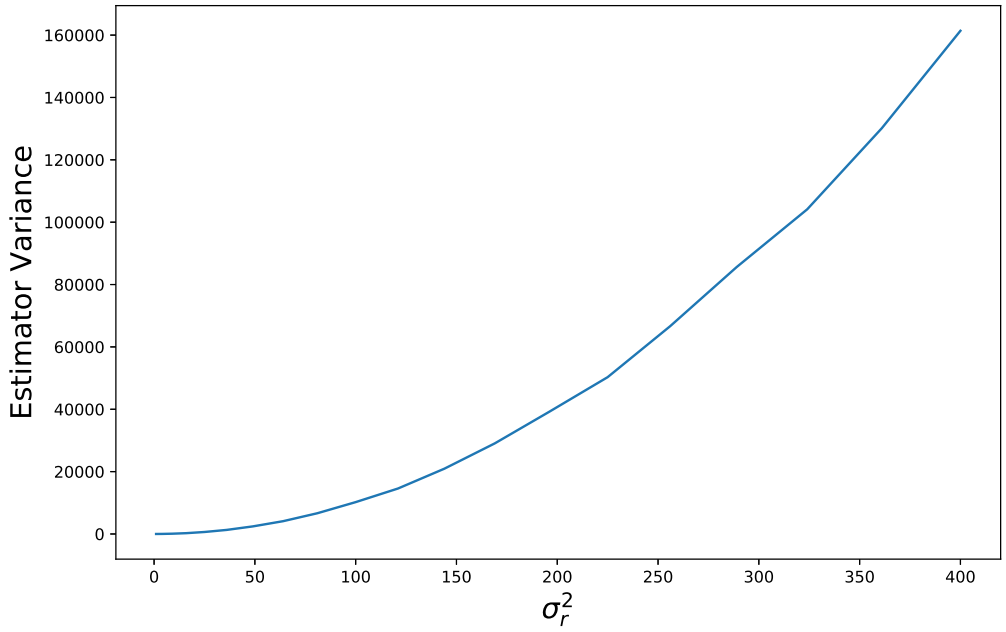

Figure 4.10: The variance of the difference between the estimate and the actual $x_1[0]$ for an increasing σ_r^2 representing the variance of the coupling weights and $x_1^{\alpha}[0]$.

Chapter 5

Privacy-Preserving Asynchronous Averaging using State Decomposition and Confidential Interaction Protocol

5.1 Introduction

The hybrid approach of state decomposition along with the confidential interaction protocol(CIP) proposes a way to deal with the coupling weight assumptions of the Chapter 4. It extends the scope of the privacy to handle the eavesdropper case with the information output given in eq. (4.15). This is done by introducing a minimum number of homomorphic encryption to the system. An eavesdropper who taps all the communication will be able to infer the initial state of the nodes if all the information shared in the system is used as is without any tricks. To achieve privacy against eavesdroppers, [24] uses secret continuous functions that are privately shared between each node. These secret functions are never exposed during the consensus process. This prevents the estimation of the initial values uniquely by an eavesdropper as there is an additional layer of security that is introduced by the functions that couple nodes. This approach, however, disrupts the scalability of the system. Every time a new node joins the system, it has to agree upon a secret function with all its neighbors using a secure channel. Instead, a fully distributed algorithm which minimizes the number of homomorphic encryptions is proposed.

CIP is a cryptographic algorithm which hides the coupling weights a_{ij} from an eavesdropper through means of additively homomorphic encryption [36]. In addition to this, it also hides the coupling weights from the nodes talking to each other as well. It is a good candidate for distributed usage as the encryption and decryption do not require any third party or an aggregator.

5.2 Methodology

The proposed algorithm lays its foundation on the methodology given in section 4.2. The consensus phase will be the same whereas, the initialization phase will be modified slightly to incorporate CIP. During the initialization phase, a node, say v_i is selected randomly to start the update. If v_i is selected for the first time, the node will decompose its state into alpha and beta states with the condition that $x_i^{\alpha}[0]+x_i^{\beta}[0] = 2x_i[0]$. The node v_i will select node v_j from the set of neighbors that have not yet gone through the CIP update. In addition to this, node v_i selects a neighboring node different from the previous node that it has gone through the initialization update, eq. (5.3), if it exists. The reason behind this choice is explained in the privacy breach in section 5.3. If node v_j is also selected for the first time, it also decomposes its state into alpha and beta states with the condition that $x_j^{\alpha}[0]+x_j^{\beta}[0] = 2x_j[0]$. The nodes keep track of which node that they have gone through the CIP update and the initialization update whichever has happened before. If the node has finished the CIP update first, it goes through the initialization update next with the node that it has not communicated during the CIP update. If it has gone through the initialization update earlier, it goes through the CIP update with a different node than the one that it has communicated earlier.

A node is said to finish the initialization process if it has gone through the CIP update, eq. (5.4), and initialization update, eq. (5.3), once. It is required, however, that these two updates are done with two different nodes to prevent the privacy breach given in section 5.3. If node v_i has only one neighboring node, it goes through the CIP update with its only neighboring node and finishes the initialization. Since there is a requirement that there are at least three nodes in the network, the other node will go through the CIP update and the initialization update once preventing the possible privacy breach that could happen.

The minimum number of encryptions in a network with N nodes is $N/2$ if N is even and $N/2+1$ if N is odd. The reason is that there should be at least one coupling weight hidden for each node. If there are even number of nodes in the network, the $N/2$ distinct pairings of these nodes will result in the minimum number of pairings. The worst case is that there will be $N-1$ number of encryptions. The algorithm 2 is designed to minimize the number of possible encryptions by forcing the nodes to select the neighboring nodes that have not gone through the CIP update already. However, there can be situations where all the neighbors of the selected node have already gone through the CIP update. In this case, the selected node has no other choice but to select one of its neighbors that have already gone through the CIP update since CIP update is a necessary condition for privacy against eavesdropper. This will increase the number of CIP updates in the system. As nodes do not know the whole network topology, forcing the number of encryptions to the minimum is not possible. It is possible, however, to minimize the number of encryptions by forcing the nodes not to select the neighboring nodes that have gone through CIP update again whenever possible.

The nodes start the consensus process if they have finished the initialization and another node in their neighborhood also finished the initialization. By starting the consensus early, it is aimed to reduce the idle time and hasten the convergence. However, this also suggests that there can be a situation where there are, say 100 nodes in the network that has started the consensus process but there are 2 nodes that still have not gone through the initialization. This can happen due to the equal probability of selecting a node that will update its state variable. The asynchronous time model lets each node to be selected in bounded time thus, all the nodes will finish initialization in bounded time

and convergence can be achieved. First, CIP will be explained before going further in details about the modified initialization phase.

5.2.1 Confidential Interaction Protocol

CIP[36] is a decentralized cryptographic method which uses additively homomorphic encryption to obtain the scaled difference of two nodes' state variables. In particular, two nodes aim to obtain the following.

$$\begin{aligned} \Delta x_{ij}[k] &= a_{ij}[k](x_j[k] - x_i[k]) \\ \Delta x_{ji}[k] &= a_{ji}[k](x_i[k] - x_j[k]) \\ \text{subject to} \quad & a_{ij}[k] = a_{ji}[k] \neq 0 \end{aligned} \tag{5.1}$$

Let there be two nodes v_i and v_j going through CIP. Their respective states x_i and x_j are assumed to be scalar. Each node have their own public and private key pair k_{p_m} and k_{s_m}, $m \in \{i, j\}$. The symmetric update weights a_{ij} and a_{ji} are created by the multiplication of two random numbers where one is generated by node v_i, $a_{i\to j}$ and the other is generated by node v_j, $a_{j\to i}$. The flow of how node v_i obtains the difference is the following.

- v_i sends $Enc(-x_i), k_{p_i}$ to v_j
- v_j encrypts x_j with k_{pi}: $Enc(x_j)$
- v_j calculates $Enc(x_j - x_i) = Enc(x_j)Enc(-x_i)$
- v_j computes $Enc(a_{j\to i}(x_j - x_i)) = Enc(x_j - x_i)^{a_{j\to i}}$ and sends it to v_i
- v_i decrypts the result: $Dec(Enc(a_{j\to i}(x_j - x_i))) = a_{j\to i}(x_j - x_i)$
- v_i multiplies the result with $a_{i\to j}$ to get the difference $\Delta x_{ij} = a_{i\to j}a_{j\to i}(x_j - x_i)$

By following the same logic, v_j also gets Δx_{ji}. There are four transmissions per round. During CIP, each node communicates with the other one twice which make the total transmission per round four. The computational bandwidth is increased because of the introduction of cryptographic operations. The encryption process requires two exponentiation which takes $\mathcal{O}(l + \log m)$ where m is the integer value of the state variable and l is the bit length of the public key. The multiplication and the modulo operations that are followed by the encryptions do not include extra computational burden compared with the exponentiation operation. In addition to this, the bit length of the public key is greater than the message m. The encryption has complexity $\mathcal{O}(l)$. The decryption has one exponentiation which is followed by arithmetic operations. Thus, decryption has complexity $\mathcal{O}(l)$. During the CIP protocol, each node v_i goes through encryption and decryption once. Assuming that bit length of the coupling weight is smaller than the bit length of the public key, the overall computational complexity of each CIP update is $\mathcal{O}(l)$.

5.2.2 Initialization Phase

Each node decomposes its state value into two substates $x_i^\alpha[0] \in \mathbb{R}$ and $x_i^\beta[0] = 2x_i[0] - x_i^\alpha[0]$. The $x_i^\alpha[k]$ states are used for the communication with the other nodes while $x_i^\beta[k]$ states are used internally. There are two updates

happening during the initialization phase: CIP update and the initialization update. The CIP update equations for the initialization phase is

$$\begin{aligned} x_i^{\alpha}[k+1] &= x_i^{\alpha}[k] + \Delta x_{ij}[k] + a_{i,\alpha\beta}[k](x_i^{\beta}[k] - x_i^{\alpha}[k])\,, \\ x_i^{\beta}[k+1] &= x_i^{\beta}[k] + a_{i,\alpha\beta}[k](x_i^{\alpha}[k] - x_i^{\beta}[k]) \end{aligned} \tag{5.2}$$

The initialization update is the non-encrypted version of the CIP update which is used in the state decomposition approach in chapter 4 and it is given as the following.

$$\begin{aligned} x_i^{\alpha}[k+1] &= x_i^{\alpha}[k] + a_{ij}[k](x_j^{\alpha}[k] - x_i^{\alpha}[k]) + a_{i,\alpha\beta}[k](x_i^{\beta}[k] - x_i^{\alpha}[k])\,, \\ x_i^{\beta}[k+1] &= x_i^{\beta}[k] + a_{i,\alpha\beta}[k](x_i^{\alpha}[k] - x_i^{\beta}[k]) \end{aligned} \tag{5.3}$$

A node is said to be initialized when it has gone through the CIP update given in eq. (5.2) and initialization update given in eq. (5.3) with two different nodes. However, there can be a situation when there is only one node in the neighborhood of the selected node. In this case, that node is said to finish initialization when it has gone through the CIP update with its only neighbor. The node is also said to finish the initialization if it has gone through two CIP updates with two different neighbors. The requirement that there is at least one CIP update is because of the protection against eavesdropper. The necessity that there is either one more CIP update or an initialization update is to prevent the privacy breach given in section 5.3.

The node v_i is selected with probability $p_i = 1/N$. It selects the node v_j from the set of neighbors that have not gone through the CIP update and different from the node that it has gone through initialization update if it exists. Since this is not a synchronous algorithm, there can be a situation where a node has gone through the initialization update before it has gone through the CIP update. As the main motivation for each node is to go through these two updates with two different nodes, they keep a track of the nodes that they have communicated for either the CIP update or the initialization update. Whenever node v_i is selected to update its state variable, it checks which nodes have not done a CIP update in its neighborhood. If it is clashing with the node that it has communicated before for the initialization update, node v_i selects that neighboring node to start a CIP update while erasing the node that it has communicated before from its memory. Minimizing the encrypted operations is given priority since one of the challenges of this thesis is to achieve the average consensus as fast as possible.

The node v_i and the node v_j start the CIP process given in section 5.2.1 and go through the update given in eq. (5.2). Node v_i writes the node v_j into memory and vice versa to prevent the selection of this node to go through the initialization update. If in this state they have already gone through the initialization update with another neighbor before, they broadcast that they have finished the initialization.

When node v_i is selected for the second time, it selects a neighboring node $v_m \neq v_j$ with equal probability. Two nodes start the initialization update given in eq. (5.3). After updating their state variables, node v_i broadcasts that it has finished the initialization. If v_m has gone through the CIP update before, it also broadcasts that it has finished the initialization.

There are at minimum $N/2$ initialization updates for even N and $(N+1)/2$ initialization update for odd N. For odd N, there will be $(N+1)/2$ number of $\mathcal{O}(l)$ operations and for even N, there will be $N/2$ number of $\mathcal{O}(l)$ operations until the initialization phase ends. At maximum, there will be $N-1$ number of $\mathcal{O}(l)$ operations. Thus, the

overall complexity level for the initialization phase is $\mathcal{O}(N \cdot l)$. The computational complexity for each node, however, does not increase with the network size and it is only dependent on the bit length of the public key, $\mathcal{O}(l)$.

Algorithm 2 Asynchronous Averaging with State Decomposition and CIP

Let $encrypt\_update(v_i)$ to return 1 if the node v_i has not gone through a CIP update and 0 otherwise.

Let $initialization\_update(v_i)$ to return 1 if the node v_i has not finished initialization and 0 otherwise.

Let F_i be the set of neighbors of v_i that have finished initialization.

The initial values are decomposed into $x[0] \in \mathrm{R}$ and $x^{\beta}[0] = 2x[0] - x^{\alpha}[0]$
for k=1,...,K **do**
    Select v_i with probability $p_i = \frac{1}{N}$
    if $encrypt\_update(v_i)$ **then**
        if $N_i \setminus F_i \neq \varnothing$ **then**
            Select $v_j \in N_i \setminus F_i$ with $p_{j|i} = \frac{1}{|N_i \setminus F_i|}$
            $v_{eav} \leftarrow V_j$
        else
            Select $v_j \in N_i$
        end if
        Get $\Delta x_{ij}^{\alpha}[k]$ using CIP
        Update using eq. (5.2)
    else if $initialization\_update(v_i)$ and $|N_i| \neq 1$ **then**
        Select $v_j \in N_i \setminus v_{eav}$
        Update using eq. (5.3)
        Broadcast initialization finished
    else if $|F_i| \neq \varnothing$ **then**
        Select $v_j \in F_i$ with probability $p_{j|i} = \frac{1}{|F_i|}$
        Send $x_i^{\alpha}[k]$
        Update using eq. (5.4)
    else
        Skip the iteration
    end if
end for

5.2.3 Consensus Phase

The nodes start the consensus phase when they have gone through initialization update and CIP update once with two different nodes and there is another node in the neighborhood that also has finished the initialization. If the node has only one neighboring node, the consensus process starts when it has gone through CIP once and the neighboring node

has finished the initialization. The update equations for the consensus process is the following.

$$\begin{aligned} x_i^{\alpha}[k+1] &= x_i^{\alpha}[k] + \frac{1}{3}(x_j^{\alpha}[k] - x_i^{\alpha}[k]) \\ &\quad + \frac{1}{3}(x_i^{\beta}[k] - x_i^{\alpha}[k])\,, \\ x_i^{\beta}[k+1] &= x_i^{\beta}[k] + \frac{1}{3}(x_i^{\alpha}[k] - x_i^{\beta}[k]) \end{aligned} \tag{5.4}$$

Node v_i that has finished initialization selects another node $v_j \in F_i$ with probability $p_{j|i} = 1/|F_i|$ where F_i represent the set of the neighboring nodes of v_i that has finished the initialization. If there are no neighbors that have finished the initialization, the selected node skips updating its value. Selected nodes go through the consensus update given in eq. (5.4) unless the stopping criterion is met.

5.3 Privacy Analysis

The privacy analysis made in section 4.3, shows that if one of the coupling weight is hidden from the adversaries, there is no range to estimate the initial values. The privacy analyses here extends this result to incorporate a way to hide the coupling weight from the adversaries using CIP. The algorithm proposed in algorithm 2, forces each node to go through at least one CIP update. The additively homomorphic encryption provides indistinguishability under chosen-plaintext attack(IND-CPA) security as explained in section 3.3.2. This means that the value which is encrypted with Paillier encryption provides IND-CPA security against the semi-honest adversaries who follow the protocol that do not maliciously change their state values. The coupling weight, say a_{ij} is hidden from all the semi-honest nodes in the system including the nodes v_i and v_j. For this reason, even though all the neighbors of the node v_j are colluding to estimate its initial value, they will not have any range to estimate it.

To show how CIP promotes a way to hide the coupling weight hidden from the nodes going through the update itself, consider the following update between node v_i and node v_j that is given in eq. (5.3). Only the part of the node v_i is given and all the values that node v_j knows is presented with an underline.

$$\begin{aligned} x_i^{\alpha}[k+1] &= x_i^{\alpha}[k] + \Delta x_{ij}[k] + a_{i,\alpha\beta}[k](x_i^{\beta}[k] - x_i^{\alpha}[k]) \\ &= x_i^{\alpha}[k] + a_{i\to j}\underline{a}_{j\to i}(\underline{x}_j^{\alpha}[k] - x_i^{\alpha}[k]) + a_{i,\alpha\beta}[k](x_i^{\beta}[k] - x_i^{\alpha}[k]) \end{aligned}$$

Even if node v_j knows $\Delta x_{ji}[k]$, there is no way that it can deduce $x_i^{\alpha}[k]$ since $a_{i\to j}$ is only known to node v_i. As there are two unknowns and one equation, there are infinitely many solutions. For this reason, the coupling weight a_{ij} and the x_i^{α} values will be hidden for node v_j with IND-CPA security. The same applies for node v_i and all the other nodes going through the CIP update. The required condition that there is one coupling weight hidden from the adversaries is satisfied with one CIP update.

The next update after the CIP update has to be an initialization update with a different node. The reason is to make sure that there are more than two nodes that have communicated with each other before starting the consensus process. The update equations preserve the sum and if two nodes start the consensus process after going through CIP update once, they will know each other's $x_i^{\beta}[k]$ and $x_j^{\beta}[k]$ values. Since the average is a deterministic function of the α and β states and they know their initial value, the other one's initial state will be disclosed. A better explanation of this

privacy breach can be explained with an example.

Privacy Breach

Each node needs to go through one CIP update and one initialization update with two different nodes unless for any v_i, $|N_i| = 1$. In this case, the initialization ends when that node goes through CIP update once. As there are always more than 2 nodes in the network and the network is connected, at least one of the nodes in the network will go through the two updates specified earlier which will prevent the privacy breach from happening.

During the consensus phase, the average of two nodes in the network is a deterministic function of the α and β states as the consensus update given in eq. (5.3) converges to the average of the initial values as shown in section 4.2.3. Let's assume that the semi-honest adversary node v_i and the node v_j has gone through CIP update once with each other. Only the iterations of node v_j will be shown.

$$x_j^\alpha[1] = x_j^\alpha[0] + \Delta x_{ji}[0] + a_{j,\alpha\beta}[0](x_j^\beta[0] - x_j^\alpha[0])$$
$$x_j^\beta[1] = x_j^\beta[0] + a_{j,\alpha\beta}[0](x_j^\alpha[0] - x_j^\beta[0])$$

In this case the sum of the α and β states of nodes v_j and v_i will not change.

$$\sum_{v=i,j} (x_v^\alpha[0] + x_v^\beta[0]) = \sum_{v=i,j} (x_v^\alpha[1] + x_v^\beta[1]) \tag{5.5}$$

Given that the next update is with node v_m that is in the neighborhood of the adversary and it is a consensus update, $x_j^\beta[1]$ will be disclosed. The update of node v_j with v_m is

$$x_j^\alpha[2] = x_j^\alpha[1] + \frac{1}{3}(x_m^\alpha[1] - x_j^\alpha[1]) + \frac{1}{3}(x_j^\beta[1] - x_j^\alpha[1])\,,$$
$$x_j^\beta[2] = x_j^\beta[1] + \frac{1}{3}(x_j^\alpha[1] - x_j^\beta[1])\,.$$

With the information output of consecutive α states

$$I_i = \{x_m^\alpha[1], x_j^\alpha[1], x_m^\alpha[2], x_j^\alpha[2], a_{m,\alpha\beta}[1] = 1/3, a_{j,\alpha\beta}[1] = 1/3\}$$

the semi-honest adversary node v_i can simply revert the consensus update that the node v_j and v_m went through to reach to the $v_j^\beta[1]$.

$$x_j^\beta[1] = 3x_j^\alpha[2] - x_j^\alpha[1] - x_m^\alpha[1]$$

The average between nodes v_i and v_j are a deterministic function of the inputs: $x_j^\beta[1]$, $x_j^\alpha[1]$, $x_i^\beta[1]$ and $x_i^\alpha[1]$ which are obtained by the semi-honest adversary in this case. Let the function returning the average of two nodes using their α and β states be defined as the following.

$$f(x_j^\beta[1], x_j^\alpha[1], x_i^\beta[1], x_i^\alpha[1]) = \frac{\frac{x_j^\alpha[1]+x_j^\beta[1]}{2} + \frac{x_i^\alpha[1]+x_i^\beta[1]}{2}}{2}$$

Using the eq. (5.5) and the function defined above,

$$f(x_j^\beta[1], x_j^\alpha[1], x_i^\beta[1], x_i^\alpha[1]) = f(x_j^\beta[0], x_j^\alpha[0], x_i^\beta[0], x_i^\alpha[0])\,.$$

Since the initial state variable, $x_i[0] = (x_i^\beta[0] + x_i^\alpha[0])/2$ is known by the semi-honest adversary and the average value between nodes v_j and v_i is disclosed, the semi-honest adversary can find the initial value of node v_j exactly.

This situation can be prevented by not fixing the coupling weights, $a_{j,\alpha\beta}[k]$ and $a_{jm}[k]$ for one more iteration after the CIP update. This will prevent the preservation of the sum of α and β states between any pair of nodes that have gone through a CIP update. Given that there is one more initialization update with a different node, say v_m, the summation between the nodes v_i and v_j will not be preserved.

$$\sum_{v=i,j}(x_v^{\alpha}[0]+x_v^{\beta}[0]) = \sum_{v=i,j}(x_v^{\alpha}[1]+x_v^{\beta}[1]) \neq \sum_{v=i,j}(x_v^{\alpha}[2]+x_v^{\beta}[2])$$

Even though the $x_j^{\beta}[2]$ state is disclosed, the average that the node v_i can get will not be the average of their initial values.

5.4 Numerical Analysis

The proposed algorithm reaches the exact average of the initial values of an arbitrary graph with more than two nodes. To be able to compare with the findings of the other chapters with this approach, a 5 node circular graph is chosen. In addition to this, a random geometric graph with 100 nodes and a radius of $\sqrt{\frac{2\log N}{N}}$ is chosen where $\log$ is in the natural base.

The random geometric graphs are commonly used in modeling wireless sensor networks[54]. The radius is chosen such that the probability of not having an isolated vertex is sufficient and given approximately by $P(isolated\_vertices) = 1-\frac{1}{N^2}$[55].

First, the convergence properties of the estimator are examined. The initial values of the nodes are selected from $[1,20]$. The coupling weights and the initial alpha state variables are selected from $[-5,5]$ during the initialization. The convergence of the alpha state variables can be seen in fig. 5.1. The stopping criterion which is the average squared distance from the true average of the initial values and the estimated average is chosen to be 10^{-9}. For both of the graph types, the proposed algorithm reaches the exact average of the initial values. The initial increase in the mean squared error is due to the initialization phase where noise is introduced to the system. The convergence rate of the consensus phase can be observed to be the same for both graphs. The overall convergence time will be different due to encrypted updates and the privacy conditions that need to be satisfied, however, the convergence rate will be the same due to its independence from the magnitude of the initial discrepancy.

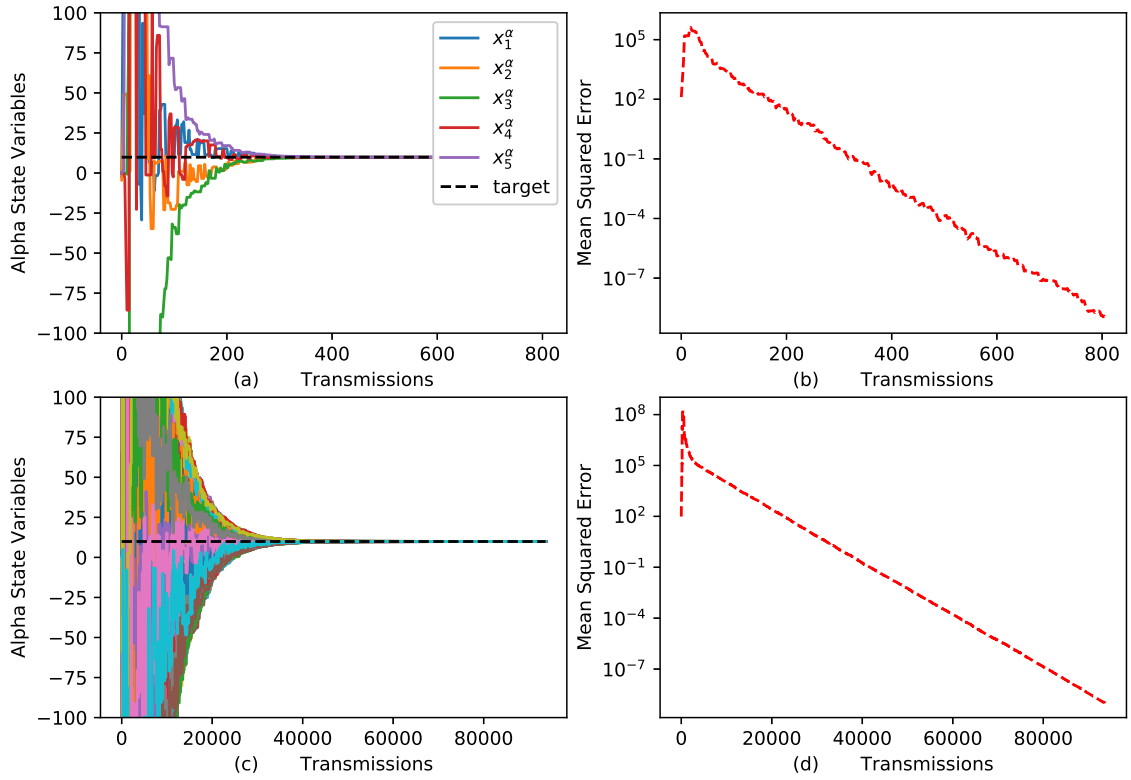

Figure 5.1: The convergence plots of the alpha states for the circular graph and the random geometric graph. The (a) and (b) represent the convergence of the alpha states and the mean squared error for the 5 node circular graph where as (c) and (d) represent the same properties for the 100 node random geometric graph

Second, an analysis on the number of the encrypted updates and the skipped iterations are done. The number of encryptions is designed to be minimized as one of the main objectives of this thesis study is to have fast and low complexity convergence. 100 simulations have been made to capture the number of encryptions and skips. Each number of the encryptions and the skips are represented as a circle in the graph. Their mean values are represented as a flat line. The findings can be seen in fig. 5.2. The mean value of the number of encryptions is found to be the minimum, 3 with a standard deviation of 0. The topological structure of the cyclic graph forces the encryptions to be the minimum as there are always 2 distinct pairings of the 5 nodes. There is an extra 1 encryption which is due to the even number of total nodes. The number of average skipped iterations is found to be 0.2 with a standard deviation of 0.6. This suggests that the uniform selection procedure of the nodes at each iteration enables enough nodes to finish initialization preventing them from skipping updates most of the time. The random geometric graph has an average of 51.51 CIP updates with a standard deviation of 0.67. For 100 nodes, the theoretical minimum of CIP updates required to preserve the privacy is given by $N/2 = 50$ encryptions. It can be seen that the algorithm tends to minimize

the number of encryption operations towards the theoretical minimum with a small deviation. The number of skipped updates are found to be similar to the cyclic graph. The average of the number of skipped updates is 0.4 with a standard deviation of 0.67. There are enough initialized nodes in the system which prevents the requirement of skipping the iterations.

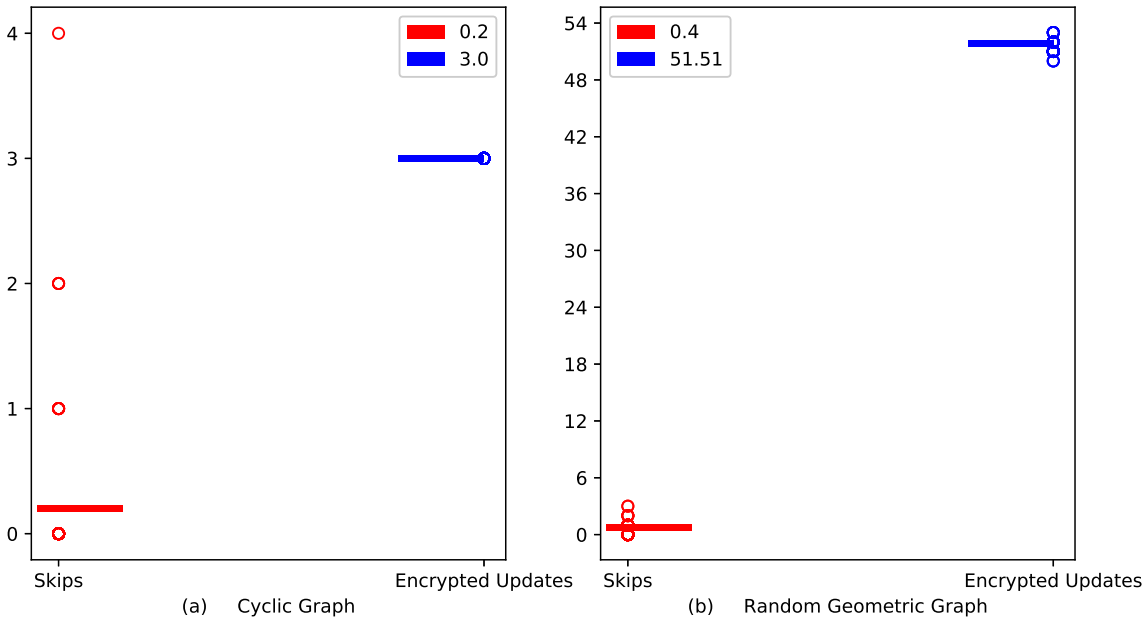

Figure 5.2: The number of CIP updates and skips repeated for 100 simulations. The (a) represent the results for the cyclic graph and the (b) represent the random geometric graph

Lastly, a convergence speed comparison is done that shows the convergence speed of the randomized gossip and the hybrid state decomposition approach. The results can be seen in fig. 5.3. The convergence rate is the same as the state decomposition approach without the CIP updates as the convergence rate is defined by the convergence phase which involves the same updates. For both of the graphs, the proposed approach has a lower convergence rate compared to the randomized gossip because of the increase in the number of nodes from N to $2N$.

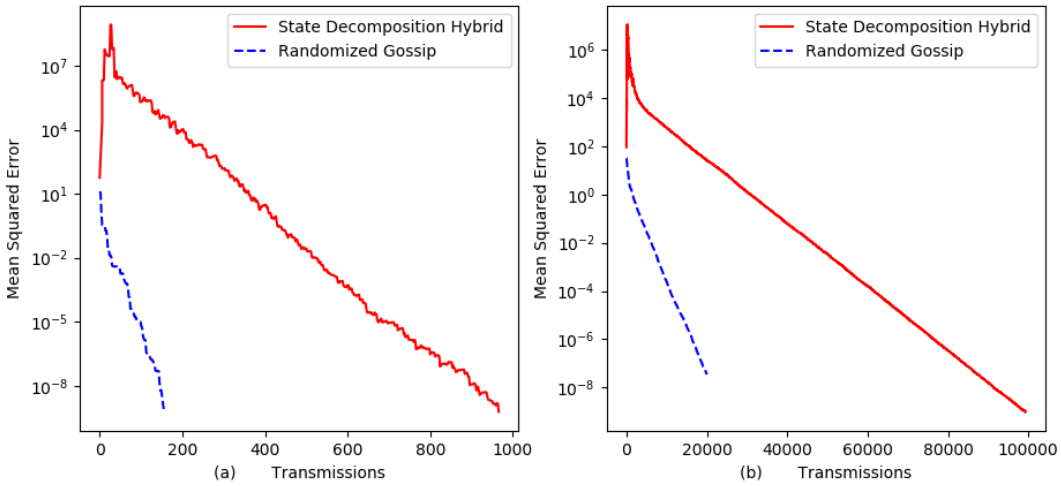

Figure 5.3: Plots of the convergence speed comparing the randomized gossip and the privacy-preserving hybrid state decomposition approach. The (a) represent the convergence speed plot for the cyclic graph and the (b) represent the convergence speed plot for the random geometric graph

Chapter 6

Privacy-Preserving Asynchronous Averaging using Noise-Obfuscation

6.1 Introduction

This algorithm aims to let N agents reach the exact average of their initial values while preserving the privacy of them from the semi-honest adversary through the addition of correlated and decaying noise. At each iteration, the nodes add noise to its state variable and average using the obfuscated values. The noise added at one iteration is subtracted in the next one and a new noise sample is added which decays by a geometric factor defined as γ. As the added noises add up to zero and decay as the iterations continue, convergence is guaranteed.

A synchronous privacy-preserving noise-obfuscation is proposed and its convergence and privacy conditions are analyzed[26]. It has been found that for each pair of nodes v_i and node v_j the following condition should hold.

$$N_i \cup v_i \not\subset N_j \cup v_j \tag{6.1}$$

This guarantees that at each synchronous update, there will be at least one unknown variable that prevents the added noises to lie in the observable space of the adversary. If all the neighbors of, say node v_i are listened by the adversary, she can find the exact initial value of the node v_i by reverting the updates that it has been going through. If there are k colluding adversaries, the condition in eq. (6.1) extends to cover all the combinations of them. For $v_1, v_2 \dots v_k \neq v_i$

$$N_i \cup v_i \not\subset N_1 \cup v_1 \cup \dots \cup N_k \cup v_1, v_2, \dots v_k$$

The asynchronous privacy-preserving noise-obfuscation method [31] analyzes the convergence rate and presents the thresholds of the maximum decay that is required for the convergence rate to be unaffected by the noise. The results suggest that if the following condition on the decay rate is satisfied, the convergence rate is not driven by noise.

$$\gamma_i \leq \sqrt{1 - \frac{\alpha(G)}{2d_i}}$$

Since the algebraic connectivity, $\alpha(G)$ might be unknown, by using the inequality $\alpha(G) \leq \frac{n}{n-1} d_{min}$, the authors

suggest the following condition on the noise decay.

$$\gamma_i \leq \sqrt{1 - \frac{(n-1)d_{min}}{2nd_i}} \tag{6.2}$$

Each node can select their decay rate, promoting a different privacy guarantee. Given that each node selects a decay parameter γ according to the eq. (6.2), the convergence is shown to be achieved. It has also been observed the convergence rate is not affected by the noise variance if the decay rate is in the given range. This information plays a critical role in the privacy proof done in section 6.3.

The authors do not give an analysis of privacy. In this chapter, the scope of their analyzes is extended to cover the privacy and numerical analysis is done.

6.2 Methodology

At every iteration a node $v_i \in N$ is selected with probability $p_i = 1/N$. The selected node contacts another node in its neighborhood, $v_j \in N_i$ with probability $p_{j|i} = 1/N_i$. For the initial iteration, the nodes add the noise $\theta_m(0)\, m \in \{i, j\}$ independently of all the other nodes. If it is not the initial iteration, the nodes generate a new noise that is generated according to the following formula.

$$w_i(k) = \begin{cases} \theta_i(0) & \text{, if } k = 0 \\ \gamma^k \theta_i(k) - \gamma^{k-1}\theta_i(k-1) & \text{, otherwise} \end{cases} \tag{6.3}$$

Each node adds $w_m(k), m \in \{i, j\}$ to their state variable and transmit it to the node they are contacting. The nodes update their state variable using the randomized gossip framework. The only difference is that nodes use the obfuscated state variables during the update.

$$x_i(k+1) = x_i^+(k) + \frac{1}{2}(x_j^+(k) - x_i^+(k)) \tag{6.4}$$

The procedure is explained in algorithm 3.

Algorithm 3 Privacy-Preserving Averaging using Noise-Obfuscation

1: **Input**: Private input values $\mathbf{x}[0] \in R^N$, noise variances $\sigma_i^2 \in R_+$ and geometric decay rate γ_i where $0 \leq \gamma_i \leq 1$ for all nodes.
2: **for** k=1,...,K **do**
3: Select v_i with probability $p_i = \frac{1}{N}$
4: Select $v_j \in N_i$ $p_{j|i} = \frac{1}{|N_i|}$
5: Generate $\theta_i[k] \sim U(-\sqrt{3}\sigma_i, \sqrt{3}\sigma_i)$ and $\theta_j[k] \sim U(-\sqrt{3}\sigma_j, \sqrt{3}\sigma_j)$
6: Set for $m \in \{i, j\}$

$$w_m(k) = \begin{cases} \theta_m[0] & \text{, if } k = 0 \\ \gamma^k \theta_m[k] - \gamma^{k-1}\theta_m[k-1] & \text{, otherwise} \end{cases} \tag{6.5}$$

7: Transmit $x_i^+[k] = x_i[k] + w_i[k]$ and $x_j^+[k] = x_j[k] + w_j[k]$
8: Update variables using eq. (6.4)
9: **end for**

6.3 Privacy Analysis

In the privacy analysis section, the motivation is to use mutual information to analyze how much information is leaked to the system with each release of the obfuscated state variables. The mutual information has been widely used as a measure of dependency between two random variables. Lower mutual information suggests independence whereas higher mutual information suggests dependence between the two random variables. Let the adversary be node v_i with the following information output.

$$I_i[k] = \{x_i^+[k], x_i[k], \theta_i(k), w_i(k), x_j^+[k]|_{j \in N_i}\} \tag{6.6}$$

The total information output of the adversary after K iterations is $I_i[\mathbf{K}] = \cup_{k=0}^{K} I_i[k]$. The adversary knows its state variables, the noise that it adds at each iteration and the obfuscated state variables that are transmitted in its neighborhood. By using $I_i[\mathbf{K}]$, the semi-honest adversary node v_i tries to gain more information about a node's initial value that is in its neighborhood.

Let the targeted node be named as node $v_j \in N_i$. The mutual information at each transmission of the node v_j can be written as the following.

$$I(x_j^+[k]; x_j[0]|I_i[\mathbf{K}])$$

Using this notations, the claim about the privacy of the initial values can be done.

Claim 2. Given the information output $I_i[\mathbf{K}]$ and the topological condition given in eq. (6.1), the adversary will not have any range to estimate the initial conditions as the noise variance goes to infinity.

Proof. Two cases need to be investigated. First one is the investigation of how much information is leaked into the system when the targeted node, v_j communicates with another node that can be intercepted by the adversary. During

the initial iteration, the noise is added to the initial value which determines how much information is leaked to the system about the initial value. Given that the noise variance goes to infinity, the release of the obfuscated state variable $x_j^+[0]$ will not have any correlation with the initial state variable, $x_j[0]$. This correlation can be seen by the following expression.

$$I(x_j^+[0]; x_j[0])$$

Define $\alpha = \frac{1}{\sigma_j^2}$ and the noise added at 0th iteration scaled by α to be $\bar{w}_j[0] = \alpha\theta(0)$ with unit variance. The mutual information will be invariant if it is scaled by α.

$$I(x_j^+[0]; x_j[0]) = I(\alpha x_j^+[0]; \alpha x_j[0])$$

As the variance of the noise goes to infinity, the mutual information will go to zero. If

$$\begin{aligned}\lim_{\sigma_j^2 \to \infty} I(\alpha x_j^+[0]; \alpha x_j[0]) &= \lim_{\alpha \to 0} I(\alpha x_j^+[0]; \alpha x_j[0]) \\ &= I(\bar{w}_j[k]; 0) = 0\end{aligned}$$

The second case is the investigation of how much information is leaked when the targeted node communicates with node, say v_m that is outside the neighborhood of the adversary. There must be one node that cannot be heard by the adversary since otherwise, the adversary will use the following equation to find the initial value.

$$s[k+1] = s[k] + x_j^+[k+1] - (x_j^+[k] + \frac{1}{2}(x_m^+[k] - x_j^+[k])) \tag{6.7}$$

The $s[0]$ is initialized as $s[0] = x_j^+[0]$ and $x_m^+[k]$ represent the obfuscated state variables of all the neighboring nodes $v_m \in N_j$. Given that $N_j \subset N_i$, it can be shown that $s[k]$ approaches to the $x_j[0]$ as the iterations go to infinity.

$$\lim_{k \to \infty} s[k] = x_j[0] + \theta(0) + \gamma\theta(1) - \theta(0) + \gamma^2\theta(2) - \gamma\theta(1) + \cdots + \gamma^k\theta(k) + \gamma^{k-1}\theta(k-1)$$

Using $0 < \gamma < 1$ and $\lim_{k\to\infty} \gamma^k = 0$,

$$\lim_{k \to \infty} s[k] = x_j[0]$$

The adversary will not be able to go through the update given at eq. (6.7), given that there is at least one node that cannot be listened by the adversary. The adversary will not know the value of $x_m^+[k] \notin N_i$. This value will blind the newly added noise to $x_j^+[k]$, represented as $w_j(k) = \gamma^k\theta_j(k) - \gamma^{k-1}\theta_j(k-1)$.

Let $T^{(1)}$ represent the iteration number where the node v_j is going through initialization update with the neighbor, say v_m that the adversary cannot listen to, for the first time. At each update with the node v_m, new unknown information is embedded into the system. At T^1th iteration, the noise $w_j[k]$ is blinded by the random variable $x_m^+[k]$. The next time the node v_j communicates v_m, a new noise sample will be blinded by the state variable of node v_m at that iteration. The accuracy of estimator reduces at each iteration when v_m is contacted and it is only dependent on the earlier iteration at which node v_m is contacted. Whenever all the other nodes are contacted, their state variables can be overheard and they can be treated as constants. The unknown information that prevents the estimator to use eq. (6.7) comes from the updates happening outside the adversary's neighborhood and it is only dependent on the earlier update. The accumulated unknown information embedded into the system forms a markov chain $x_j[0] \to x_j[T^{(1)}] \to x_j[T^{(2)}] \to \cdots \to x_j[T^{(\infty)}]$. By using data processing inequality, for any iteration $T' > T^{(1)}$ it

can be said that

$$I(x_j^+[T']; x_j[0]|I_i[\mathbf{T'}-\mathbf{1}]) \leq I(x_j^+[T^{(1)}]; x_j[0]|I_i[\mathbf{T^{(1)}}-\mathbf{1}])\,. \tag{6.8}$$

To estimate the initial value of node v_j, she has to estimate the added noises at each iteration. The information leakage about the noise that the adversary tries to get at the iteration $T^{(1)}-1$ can be written as

$$\begin{aligned}
I(x_j^+[T^{(1)}]; x_j[0]|I_i[\mathbf{T^{(1)}}-\mathbf{1}]) &= I(x_j^+[T^{(1)}]; w_j[T^{(1)}-1]|I_i[\mathbf{T^{(1)}}-\mathbf{1}]) \\
&= I\Big(x_j^+[T^{(1)}]; x_j^+[T^{(1)}] - (x_j^+[T^{(1)}-1] + \frac{1}{2}(x_m^+[T^{(1)}-1] - x_j^+[T^{(1)}-1])|I_i[\mathbf{T^{(1)}}-\mathbf{1}]\Big) \\
&= I\Big(x_j^+[T^{(1)}]; x_j^+[T^{(1)}] - (\frac{x_j^+[T^{(1)}-1]}{2} + \frac{x_m^+[T^{(1)}-1]}{2})|I_i[\mathbf{T^{(1)}}-\mathbf{1}]\Big)\,.
\end{aligned}$$

Since this is a markov chain, the conditional information $I_i[\mathbf{T^{(1)}}-\mathbf{1}]$ which is the accumulation of all the transmitted states of node v_j can be reduced to $I_i[T^{(1)}-1]$ as the latest iteration's information output is enough to represent all the earlier iterations. In the contents of $I_i[T^{(1)}-1]$, only $x_j^+[T^{(1)}-1]$ affects the mutual information.

$$\begin{aligned}
I(x_j^+[T^{(1)}]; x_j[0]|I_i[\mathbf{T^{(1)}}]) &= I\Big(x_j^+[T^{(1)}]; x_j^+[T^{(1)}] - (\frac{x_j^+[T^{(1)}-1]}{2} + \frac{x_m^+[T^{(1)}-1]}{2})|I_i[\mathbf{T^{(1)}}-\mathbf{1}\Big) \\
&= I\Big(x_j^+[T^{(1)}]; x_j^+[T^{(1)}] - (\frac{x_j^+[T^{(1)}-1]}{2} + \frac{x_m^+[T^{(1)}-1]}{2})|x_j^+[T^{(1)}-1]\Big)
\end{aligned}$$

Define $\alpha = \frac{1}{\sigma_j^2}$ where σ_j^2 and $\beta = \frac{1}{\sigma_m^2}$ where σ_j^2 represent the noise variance of the node v_j and σ_m^2 represent the noise variance of the node v_m. Since $x_j^+[T^{(1)}-1]$ is given, it can be treated as a constant and it will not change the mutual information.

$$I\Big(x_j^+[T^{(1)}]; x_j^+[T^{(1)}] - (\frac{x_j^+[T^{(1)}-1]}{2} + \frac{x_m^+[T^{(1)}-1]}{2})|x_j^+[T^{(1)}-1]\Big) = I\Big(x_j^+[T^{(1)}]; x_j^+[T^{(1)}] - \frac{x_m^+[T^{(1)}-1]}{2}\Big)$$

Let $\bar{w}_j(T) = \alpha(\gamma^T\theta_j(T) - \gamma^{T-1}\theta_j(T-1))$ which creates a sum of scaled unit variance noise distributions, $\theta_j(T)$.
Let $\bar{w}_m(T) = \beta(\gamma^T\theta_j(T) - \gamma^{T-1}\theta_j(T-1))$ which creates a sum of scaled unit variance noise distributions, $\theta_m(T)$.
The mutual information is invariant to scaling. Using this property, the mutual information can be rewritten as,

$$\begin{aligned}
I\Big(x_j^+[T^{(1)}]; x_j^+[T^{(1)}] - \frac{x_m^+[T^{(1)}-1]}{2}\Big) &= I\Big(x_j[T^{(1)}] + w_j[T^{(1)}]; x_j[T^{(1)}] + w_j[T^{(1)}] - \frac{x_m[T^{(1)}-1] + w_m[T^{(1)}-1]}{2}\Big) \\
&= I\Big(\alpha\beta(x_j[T^{(1)}] + w_j[T^{(1)}]); \alpha\beta(x_j[T^{(1)}] + w_j[T^{(1)}] - \\
&\qquad \frac{x_m[T^{(1)}-1] + w_m[T^{(1)}-1]}{2})\Big) \\
&= I\Big(\alpha\beta(x_j[T^{(1)}] + w_j[T^{(1)}]); \alpha\beta(x_j[T^{(1)}] + w_j[T^{(1)}] - \\
&\qquad \frac{x_m[T^{(1)}-1] + w_m[T^{(1)}-1]}{2})\Big) \\
&= I\Big(\alpha\beta x_j[T^{(1)}] + \beta\bar{w}_j[T^{(1)}]); \alpha\beta x_j[T^{(1)}] + \beta\bar{w}_j[T^{(1)}] - \\
&\qquad \frac{\alpha\beta x_m[T^{(1)}-1] + \alpha\bar{w}_m[T^{(1)}-1]}{2}\Big)
\end{aligned}$$

As the noise variances go to infinity, it can be shown that the mutual information given above will go to zero for a

bounded geometric decay rate between 0 and 1.

$$\lim_{\substack{\sigma_j^2\to\infty\\ \sigma_m^2\to\infty}} I\Big(\alpha\beta x_j[T^{(1)}]+\beta\bar{w}_j[T^{(1)}];\alpha\beta x_j[T^{(1)}]+\beta\bar{w}_j[T^{(1)}]-\frac{\alpha\beta x_m[T^{(1)}-1]+\alpha\bar{w}_m[T^{(1)}-1]}{2}\Big)=$$

$$\lim_{\substack{\alpha\to 0\\ \beta\to 0}} I\Big(\alpha\beta x_j[T^{(1)}]+\beta\bar{w}_j[T^{(1)}];\alpha\beta x_j[T^{(1)}]+\beta\bar{w}_j[T^{(1)}]-\frac{\alpha\beta x_m[T^{(1)}-1]+\alpha\bar{w}_m[T^{(1)}-1]}{2}\Big)=I(0;0)$$

This mutual information is shown to be smaller than the mutual information in the next iterations as shown in eq. (6.8). Using the fact that the mutual information between the release of the new obfuscated states and the noise variance at the earlier state goes to zero with increasing variance, the following can be shown.

$$0\leq I(x_j^+[T'];x_j[0]|I_i[\mathbf{T'}-\mathbf{1}])\leq I(x_j^+[T^{(1)}];x_j[0]|I_i[\mathbf{T^{(1)}}-1])=0$$

The release of the obfuscated state at the $T^{(1)}$th iteration, will not reveal any information about the noise variance that is added at $T^{(1)}-1$st iteration with the information output $I_i[\mathbf{T^{(1)}}-1]$. This noise value is necessary for the adversary to estimate the initial value of the targeted node v_j. It has been shown that the release of the new obfuscated state does not have any information about the noise variance at the earlier iteration if the noise variances go to infinity. In the first case, it has been seen that there is no information leakage about the initial state if the noise variances go to infinity. The first case makes sure that the initial value is blinded enough within the neighborhood of the adversary and second case makes sure that the adversary can not estimate one of the noise values that would have been used in estimation. Both cases are satisfied and there is no correlation with the obfuscated state variable at $x_j^+[T^{(1)}]$th iteration and the initial value $x_j^+[0]$ of node v_j. Any iteration afterwards can only reduce the information that can be gained to deduce the initial value. Since there is no dependency between $x_j^+[T^{(1)}]$th iteration and $x_j^+[0]$, no information can be gained afterwards.

□

6.4 Numerical Results

The proposed algorithm reaches to the exact average of the initial values of the nodes of the graph G with more than 2 nodes. For the privacy to be established it needs to be connected according to the topological assumption given in eq. (6.1). The following graph topology is selected which guarantees that any pair of nodes do not share the same neighboring nodes.

The convergence properties of the proposed algorithm can be seen in the fig. 6.2. The initial values are selected to be between the range $[1, 20]$ and the noise variance is selected to be 100. The geometric decay rate γ is selected to be $0.9 < \gamma = 0.98$ which guarantees the convergence rate not to be driven by the noise variance. The algorithm converges to the exact mean of the initial values.

The convergence rate of the proposed algorithm for four different variances are plotted in the fig. 6.3. To have less irregular results, the number of nodes in this analysis is increased from 5 to 20 while keeping the cyclic topology the same. It can be seen that the convergence rate is the same as the standard randomized gossip approach regardless of the noise variances. The overall convergence time, however, increases with the increased noise variance due to the

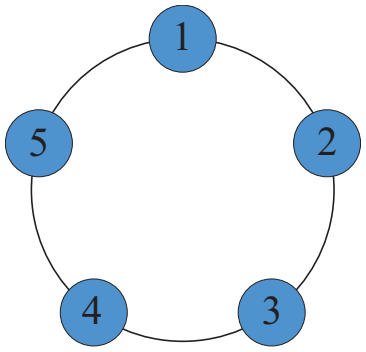

Figure 6.1: The graph topology

addition of noise components.

An experiment is done to show the effect of increasing variance on mutual information. The initial state values are sampled from a uniform distribution with unit variance. The noise variance θ_r which is fixed for all the nodes in the system is sampled from uniform distribution with the variances of $\sigma_r^2 = \{1, 4, 9, 16, 25, 36, 49, \cdots, 400\}$. The geometric decay rate is chosen to be 0.9. Node v_j is assumed to be the targeted node and node v_m is assumed to be the node that the adversary cannot listen to. It has been assumed that the first iteration that the node v_j is going through is with the node v_m. The goal is to show that the following mutual information goes to zero with the increasing noise variances.

$$I(x_j^+[1]; x_j^+[1] - (\frac{x_j^+[0]}{2} + \frac{x_m^+[0]}{2})|x_j^+[0])) = I(x_j^+[1]; x_j^+[1] - \frac{x_m^+[0]}{2})$$

The topological assumption eq. (6.1) suggests that at each update that is happening with the node v_m will improve the privacy since each update inserts independent noise to the system. The data processing inequality as shown in section 6.3 suggests that more information cannot be gained by the post-processing. The mutual information goes to zero with increasing noise variance as shown in fig. 6.4. The iterations afterward will not increase the mutual information thus, there will be no range to estimate the initial values. It can be seen that with even one update that can not be heard by the adversary can force mutual information to be arbitrarily small. The other updates will improve the privacy for a bounded noise variance however, the privacy can already be established with one-shot perturbation.

Lastly, an analysis of the estimator's performance is done. The initial values are selected from a uniform distribution with unit variance, $U[-\sqrt{3}, \sqrt{3}]$. The adversary is defined to be the semi-honest node v_i eavesdropping on the communications of node v_j. The information output of the adversary is given in eq. (6.6). Whenever node v_j is communicating with the nodes that can be listened by the node v_i, the adversary goes through the update given in eq. (6.7) to accumulate the noises which cancel out each other as the iterations go to infinity. However, whenever the node v_j communicates the node that is not in the neighborhood of the adversary which can be called as node v_m, the adversary has to guess the noise. The noise is sampled from uniform distributions and it is in the form of

$$w_m(k) = \begin{cases} \theta_j[0] & \text{, if } k = 0 \\ \gamma^k \theta_j[k] - \gamma^{k-1}\theta_j[k-1] & \text{, otherwise .} \end{cases}$$

Since the adversary is assumed to be keeping track of all the communications of node v_j, it can be assumed that it

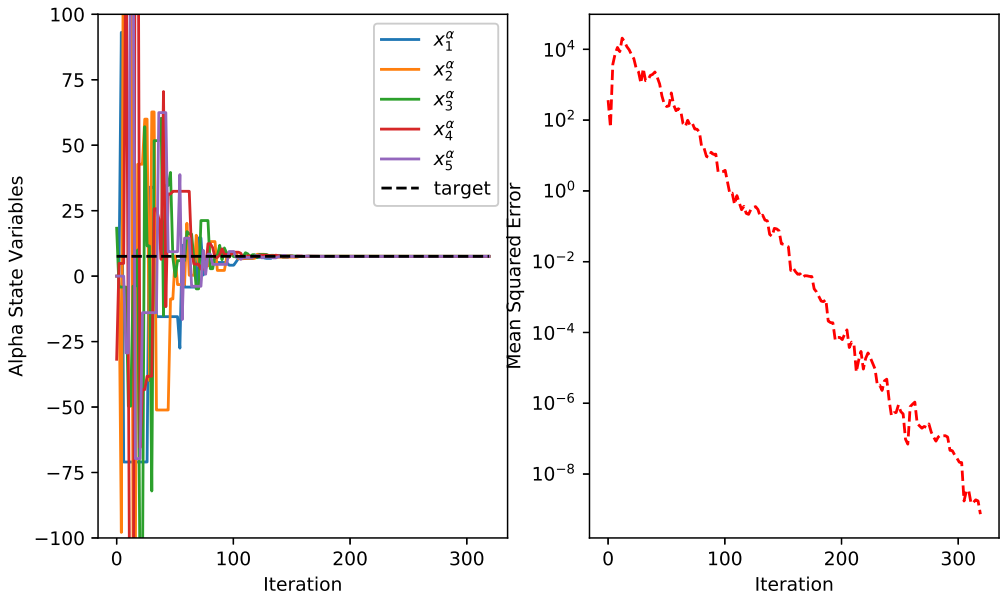

Figure 6.2: Convergence for 5 node cyclic graph

knows the iterations at which the node is currently at. This will reveal the decay rate γ^k which can be used to estimate the noise. The nodes sample their noises, $\theta[k]$ from uniform distributions with variance σ_r^2. The uniform distribution has mean 0 and symmetric around 0. In terms of the variance σ_r^2, the noise takes values in the range $[-\sqrt{3}\sigma_r, \sqrt{3}\sigma_r]$ with equal probability. If the node v_j communicates with node v_m that cannot be listened by the adversary at 0th iteration, the adversary estimates the noise as a random number between $[-\sqrt{3}\sigma_r$ and $\sqrt{3}\sigma_r]$. If the iteration is greater than 0, it estimates a random number between $-\gamma^k\sigma_r\sqrt{3}$ and $\gamma^k\sigma_r\sqrt{3}$ because it is the maximum likelihood of the noise distribution. The noise distribution for $k > 0$ is given by the expression $\gamma^k\theta_j[k] - \gamma^{k-1}\theta_j[k-1]$. In terms of the noise variance σ_r^2, the probability distribution of this expression is given in fig. 6.5. The scaling with γ^k scales the variance of the uniform distribution to be $\gamma^{2k}\sigma_r^2$. The sum of the two distributions scaled by γ^{k-1} and γ^k will result in the distribution given in fig. 6.5. For any $k > 0$, the adversary guesses a random number in the range $[-\gamma^k\sigma_r\sqrt{3}, \gamma^k\sigma_r\sqrt{3}]$ with equal probability since noise resides in this range with maximum probability.

There has been $N = 10^4$ simulations where the estimator guesses the initial value of node v_j. At every talk with the node v_m that cannot be heard by the adversary, the adverserial node v_i guesses the noise according to the following rule.

$$\hat{w}_j[k] = \begin{cases} c \sim U[-\sqrt{3}\sigma_r, \sqrt{3}\sigma_r] & \text{, if } k = 0 \\ c \sim U[-\gamma^k\sigma_r\sqrt{3}, \gamma^k\sigma_r\sqrt{3}] & \text{, if } k > 0\,. \end{cases}$$

To estimate the variance of the adversary estimates, the unbiased estimator with bessel correction [52] is used.

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_j[0] - \hat{x}_j[0])^2$$

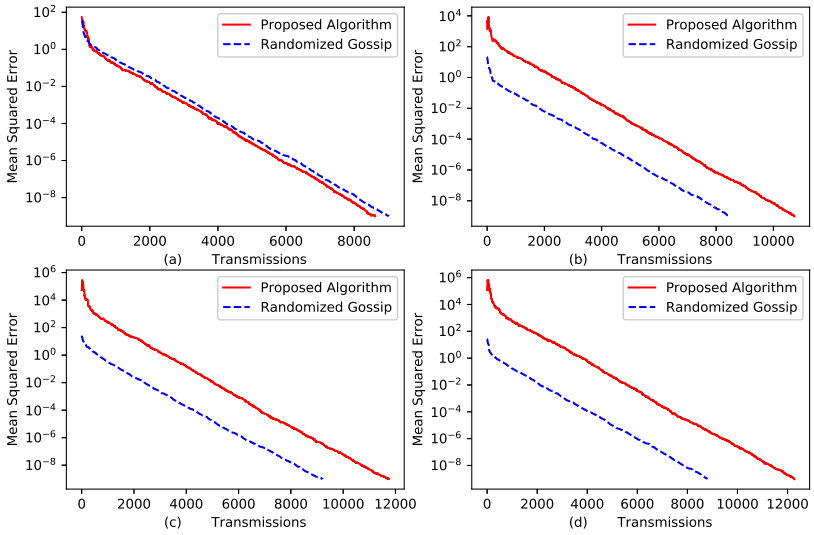

Figure 6.3: The convergence rate graphs for the noise-obfuscation method and standard randomized gossip. (a),(b ,(c) and (d) represent the noise variances $\sigma^2 = \{1, 100, 500, 1000\}$ respectively.

where s^2 represent the variance of the estimates. The results can be seen in fig. 6.6. There is a linear relationship between the variance of the estimator guesses and the increasing variance of the noise. The reason is the linearity of the additive noise. Whenever the node v_m that cannot be listened by the adversary is contacted, the estimator's performance worsens linearly with respect to the noise variance σ_r^2 because the obfuscated state variables are blinded with the additive noise. The variance of the blinded state variables, $x_j^+[k]$, is the sum of the noise variance and the state variable. For this reason, the linearity can be observed at the variance of the adversary's estimate.

To understand the effect of varying decay rate, γ, the same experiment where the adversary estimates the initial value of the node v_j is done. The decay rate, γ takes the values between 0.5 and 0.9. The estimator's performance is shown in the fig. 6.7. It can be seen that there is a geometric rate of increase in the estimator variance with the increasing γ. In the privacy section, it has been proved that for a bounded γ which prevents the convergence to be unaffected by the noise variance, privacy improves with the increasing noise variance. In addition to this, it is shown here that the increasing γ, also improves the privacy of the initial values. The reason is that at each iteration with the node v_m that cannot be heard by the adversary, new unknown information is embedded into the system whose variance depends on the magnitude of the γ. With a high γ, the noise variance that blinds the state variables will be high and that will cause an improvement in the noise variance.

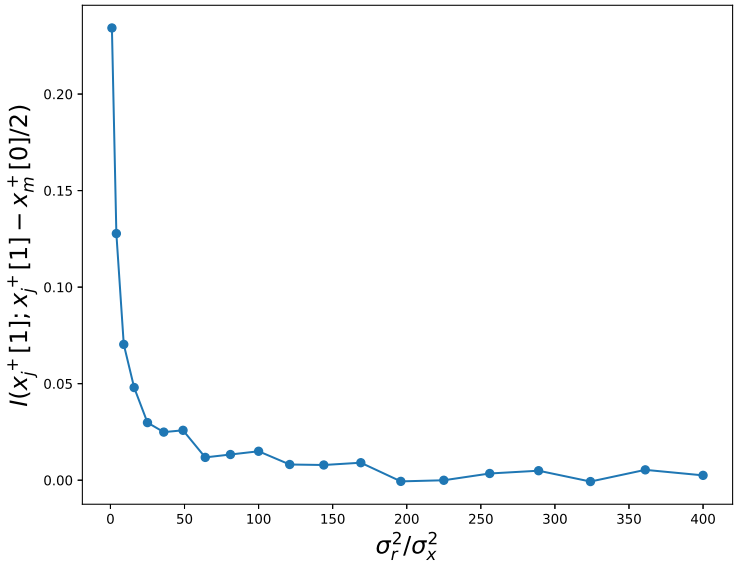

Figure 6.4: Mutual information plot for $I(x_j^+[1]; x_j^+[1] - \frac{x_m^+[0]}{2})$ with increasing noise variance

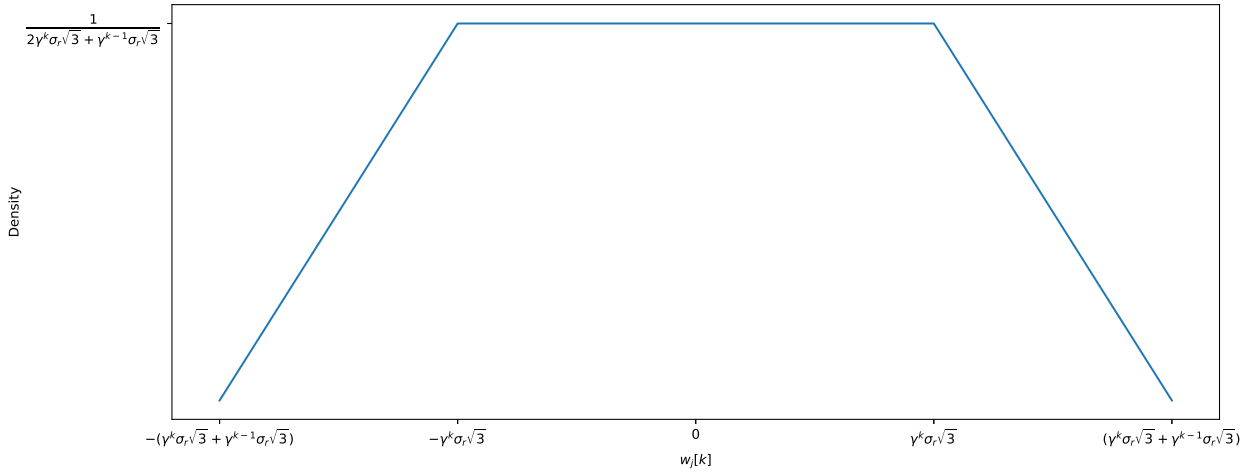

Figure 6.5: The pdf of the noise to be estimated for $k > 0$

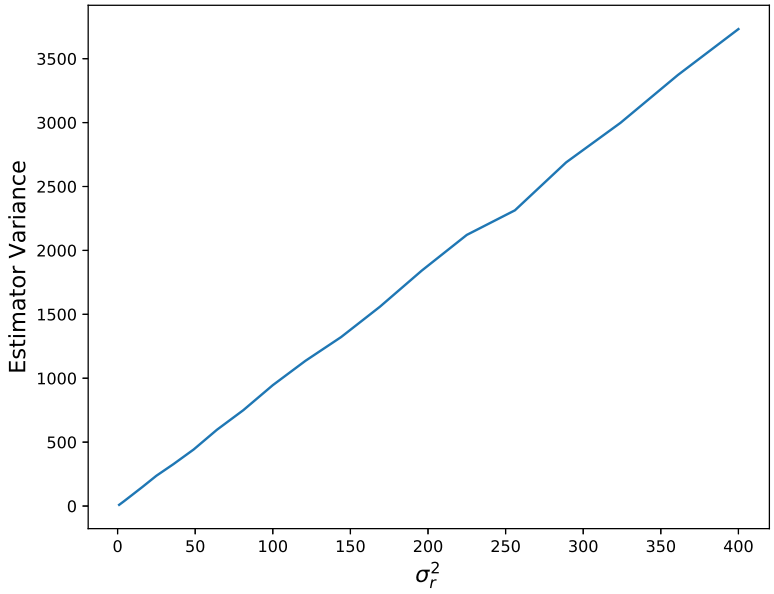

Figure 6.6: The variance of the adversary's estimates for different noise variance

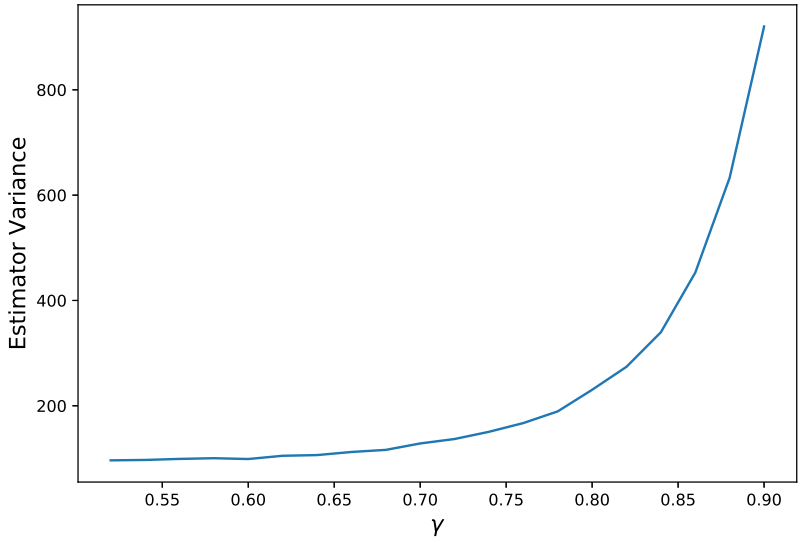

Figure 6.7: The variance of the adversary's estimate for different geometric decay rate γ

Chapter 7

Results and Future Work

There have been three methods proposed to solve the problem of achieving the exact average consensus while preserving the privacy of the initial values. Each method tackles three main challenges which are also given in the chapter 2. These are

- The nodes will reach to the average of their initial values, $x_{\text{ave}} = \sum_{j=1}^{N} x_j[0]$ through asynchronous updates.
- Each nodes initial value will be hidden from the adversaries throughout the process.
- The convergence time and computational complexity should be reduced compared to cryptographic solutions.

Firstly, an analysis of the proposed algorithms will be done with regards to the three challenges proposed earlier. Secondly, an overall analysis of the limitations and challenges of the privacy-preserving average consensus task will be done. Recommendations for future work are provided whenever a possible area of research is observed.

7.1 Analysis of Privacy-Preserving Consensus Averaging via State Decomposition

The introduction, methodology and numerical examples regarding the privacy-preserving consensus averaging via state decomposition is given in the chapter 4. The algorithm tackles the consensus averaging problem while giving a privacy guarantee to the initial values through decomposing the state variable into two states namely, $x_i^{\alpha}[0] \in \mathbb{R}$ and $x_i^{\beta}[0] = 2x_i[0] - x_i^{\alpha}[0]$. Instead of solving the average consensus problem, the algorithm transforms the problem into a constrained optimization problem with the constraints that $x_i^{\alpha} = x_j^{\alpha}$ for every node $v_i \neq v_j$ and $x_i^{\beta} = x_i^{\alpha}$ for all nodes $v_i \in N$.

Due to the state decomposition, the number of nodes in the network increases from N to $2N$. This causes a reduction in the convergence rate. However, it is possible to fix the convergence rate to be the same as the standard randomized gossip approach. As the privacy is already established during the initialization phase and the β state is disclosed during the consensus phase, the nodes who have finished initialization may merge their α and β states

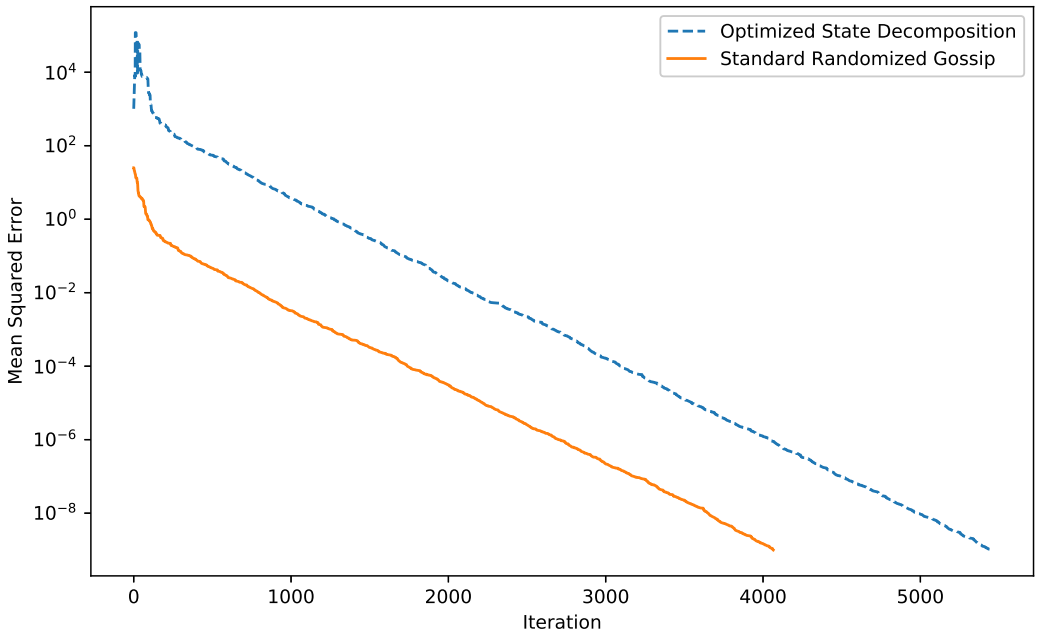

Figure 7.1: Convergence rate for standard randomized gossip and the optimized state decomposition approach

according to the following formula, $x_i[k] = (x_i^{\alpha}[k]+x_i^{\beta}[k])/2$. Nodes who have finished initialization will merge their state variables into one and start the consensus phase with the other nodes that also have finished the initialization. This will improve the convergence rate to be the same as the standard randomized gossip algorithm. A convergence test is done using a 20 node cyclic graph with $x[0] \in [1, 20]$, $x_i^{\alpha}[0], a_{i,\alpha\beta}, a_{ij}[0] \in [-5, 5]$ for $v_i \in N, v_j \in N, v_i \neq v_j$. It can be seen in fig. 7.1 that the convergence rate can be optimized to be the same as the standard randomized gossip algorithm if the nodes who have finished initialization fix their new state variable to be $(x_i^{\alpha}[k] + x_i^{\beta}[k])/2$. In the optimized version, whenever a new node joins the system, its neighboring nodes need to decompose their state variables into two while keeping the mean of the α and β states to be its state variable at that iteration. The new node has to go through the initialization update with the neighboring nodes before merging their nodes into one again. This does not disrupt the scalability of the algorithm and can still preserve the privacy of the initial values.

The $x_i^{\alpha}[k]$ state variables are used for communicating with the other nodes while $x_i^{\beta}[k]$ state variables are kept as a secret. It has been found in the analysis that if one of the coupling weight is hidden from the semi-honest adversary in the system, the adversary has no range to estimate the initial values with any accuracy as the variances of the coupling weight go to infinity. It is found, however, if $x_i^{\alpha}[0]$ states are sampled from an infinitely large variance random variable, the variance of the adversary's estimate increases exponentially. If the assumption which is given in eq. (4.4) is assumed, this will force also the $x_j^{\alpha}[k]$ to be hidden from the adversary. Using the same assumption, it has been found that the variance of the adversary's estimates in noise-obfuscation technique increases linearly with

the noise variance. The reason why state decomposition promises an exponential increase in the adversary's estimate is because of the multiplications with the hidden coupling weights $a_{j,\alpha\beta}[k]$ and $a_{jm}[k]$. These are sampled at each iteration during the initialization and they are independent of the α and β state variables which result in a random variable whose variance scales exponentially with the increasing variance of the α state variables and the coupling weights.

The problem is analyzed at a graph $G = (V, E)$ where the number of nodes, $N > 2$. In addition to this, it has been assumed that there are no channel encryptions in the system which enables the nodes to capture the transmitted messages in their neighborhood. This approach has been followed in the works [26][24][25][31] and it has been followed also in this work. In all of the non-cryptographic methods that preserve the privacy of the initial values except [24], it has been found that if the adversary listens to all the neighbors of the targeted node, the initial value of the targeted node will be disclosed. There are several ways of handling this assumption which is discussed in the section 7.5. In the state decomposition approach, a theoretical proof has been given without providing a way to hide the initial value from the adversary. It has been shown however the topological assumption eq. (4.4) which is also used in the noise obfuscation method can be used as a way to hide the one of the coupling weight $a_{jm}[k]$ and the alpha state $x_m^{\alpha}[k]$.

The state decomposition approach provides security against a semi-honest adversary in the system given that one of the coupling weight between the targeted node and its neighbor is hidden from the adversary. As the variances of the coupling weight go to infinity, there is no range to estimate the initial value with any guaranteed accuracy. If the adversary is defined to be an eavesdropper who is tapping all the communication channels in the network, the initial value of the targeted node will be disclosed. He et. al.[24] provided a way to extend the scope of the privacy to cover the eavesdropper case without using any encryptions through the inclusion of secret continuous function that is predefined for each pair of node and different among the different pairs of nodes. These secret continuous functions are arranged such that the sum of the states never change such that the exact consensus can be achieved. This, however, disrupts the scalability of the system as each pair of node agrees on a secret function through a secure channel. This secure channel can be impractical to be created when a new node wants to be included in the system. The new node has to find a way to agree on a function with its neighboring nodes using a secure channel. Instead, a scalable and distributed encryption technique[36] is used along with the state decomposition method to extend the scope of the privacy to cover the eavesdropper case. The hybrid state decomposition approach is designed to use the minimum number of encryptions to provide privacy against a semi-honest adversary and an eavesdropper.

7.2 Analysis of Privacy-Preserving Average Consensus via Hybrid State Decomposition

The hybrid state decomposition approach lays its foundation on the results given in chapter 4. The update equations are pretty similar with only differences being the singular inclusion of an encrypted operation for each node and the number of initialization updates. In the state decomposition approach, the algorithm is designed to incorporate $|N_i|$

initialization updates for each node $v_i \in N$ to make sure that each node communicates with the node that cannot be heard by the adversary. Although it has been assumed that one of the coupling weight is hidden from the adversary, there is no assumption on which one it is. This creates the necessity to traverse all the nodes to make sure one initialization update is done with each neighboring node before starting the consensus phase.

In the hybrid state decomposition's privacy chapter given in section 5.3, it has been shown that two initialization updates for each node are enough to achieve privacy. One of these initialization updates is an encrypted update which achieves IND-CPA security. The other non-encrypted initialization update needs to be done with a different node than the one before. Since the convergence phase releases the α and β state variables through fixing the coupling weights and the convergence is a deterministic function of these state variables, two nodes will be able to learn the other one's initial value. For this reason, each node is forced to go through two initialization updates where at least one is a cryptographic update.

The same optimization algorithm for state decomposition which resulted in the plot of fig. 7.1 can also be applied here. The nodes who have finished initialization will merge their state variables into one and start the consensus phase with the standard randomized gossip. Using this optimization, the convergence rate will be the same as the convergence rate of the standard randomized gossip algorithm.

It has been shown that if one of the coupling weight is hidden from the adversary, privacy can be achieved. The encrypted update CIP explained in section 5.2.1, enables the nodes going through the encrypted initialization update to learn the scaled differences of each other's α state variable. The semi-honest adversary or the eavesdropper will not be able to learn the coupling weight $a_{jm}[k]$ or the α state variables. In addition to this, the nodes also can not learn the α state variable and the coupling weight that they are sharing. With one CIP update, the coupling weight shown to be hidden from the adversary for the nodes that are involved in the update.

The CIP update lets the algorithm to be private against an eavesdropper and the semi-honest adversary. In the literature[24], secret continuous functions have been used to make the algorithm private against the eavesdropper. The pre-defined functions which couple the nodes disrupt the scalability of the system. Each new node that joins the network needs to agree on a function using a secure channel or a third party's involvement. CIP, on the other hand, is a distributed algorithm which lets the nodes to use homomorphic encryption to make the system private against an eavesdropper. In graphs where the nodes are designed to be fixed, the nodes can agree on a pre-defined function, computational power is scarce and there is no possibility to do a pre-processing, the secret continuous functions can be put to use. On the other hand, if the system needs to be scaled and a fully distributed algorithm wants to be designed without an assumption on third parties or secure channels, a single CIP operation can be incorporated. For each node, only one CIP update needs to be done. This can be thought of as a pre-processing step which does not affect the convergence rate but increases the overall convergence time.

7.3 Analysis of Privacy-Preserving Average Consensus via Noise-Obfuscation

The privacy-preserving noise-obfuscation method achieves privacy through the addition of correlated and decaying noise. At each iteration, the state variable is blinded by the addition of a scaled new noise sample and the subtraction

of the earlier iteration's scaled sample. Since the algorithm is designed to converge as the scaling factor becomes sufficiently large, a new noise has to be sampled at each iteration.

The privacy against the semi-honest adversary assumes the topological condition given in eq. (6.1) which enables one of the noise samples to be hidden from the adversary. At each iteration happening with the node that cannot be listened by the adversary, new unknown information is embedded into the system. During the privacy analysis, it has been shown that for a bounded γ that makes the convergence to be unaffected by the noise variance, the privacy can be established in one iteration. As the noise variance goes to infinity, the semi-honest adversary will not have any range to estimate the initial value with any guaranteed accuracy with one-shot perturbation. If the adversary can listen to all the communications happening between the target node and its neighbors, the initial value of the targeted node will be disclosed. For this reason, the algorithm is not private against an eavesdropper.

If the variance of the adversary's estimates is analyzed, it can be observed that the variance increases linearly with the increasing noise variance. At each iteration happening with the node that cannot be listened by the adversary, worsens the performance of the adversary in a linear fashion. The reason is the usage of the additive noise. The new noise sample that is added at each iteration blinds the state variable using an additive noise. This causes the variance of the adversary's estimate affects linearly with respect to the noise variance. The noise decay rate γ, on the other hand, has an exponential effect on the adversary's performance. As γ increases, the variance of the adversary's estimate increases. The noise decay rate γ scales the variance of the noise to be scaled by γ^2. For this reason, there is an exponential relation with the γ whereas there is a linear relation with the noise variance. The results in the numerical analysis suggest that for a bounded noise variance, a higher γ will result in a better privacy guarantee.

The work on asynchronous noise-obfuscation techniques [31] suggests that if the noise decay rate γ is chosen according to the rule given in eq. (6.2), the convergence rate will be independent of the magnitude of the noise variance. This enables the privacy proof given in section 6.3 to be valid which assumes analyzes the asymptotic properties of the mutual information as the noise variance goes to infinity. As the convergence can still be achieved when the variance of the noise goes to infinity, observing the mutual information in this setting does not disrupt the convergence property of the algorithm.

7.4 Discussion on Privacy

The privacy-preserving average consensus algorithms that reach to the exact average of the initial values require information of one of the neighboring nodes of the targeted node to be hidden from the adversary[26]. There have been several methods in the literature proposed to hide this information from the adversary. One approach[26][24][25][31] assumes that there are no channel encryptions in the network which make all the transmitted messages in the system susceptible to eavesdropping. In this approach, the information between the node and one its neighbor is designed to be protected via arranging the radius of communication. The following graph topology which is also presented in the chapter 4 and chapter 6 is an example of this approach.

$$N_i \cup v_i \not\subset N_j \cup v_j$$

It is assumed that there is only one semi-honest adversary in the system and it targets one of its neighboring nodes. If there are k colluding semi-honest adversaries in the system, the topological assumption is arranged such that it covers all the k semi-honest adversaries.

$$N_i \cup v_i \not\subset N_1 \cup v_1 \cup \ldots \cup N_k \cup v_1, v_2, \ldots v_k$$

The circular graph is an example that satisfies this graph topology. The radius of transmission can be arranged such that there is always one neighboring node in the graph that any pair of nodes do not share.

Another approach is to assume the existence of channel encryptions. In this setup, the nodes are distributed into honest and semi-honest nodes. The semi-honest nodes are assumed to be colluding which means that they share information to find out the initial value of the targeted node. Assuming that one of the neighbors of the targeted node is an honest node who does not collude with the semi-honest nodes, gives a way to hide the information of one of the targeted node's neighbors from the adversaries. The adversaries in this case, do not listen to the communications happening in their neighborhood but share the information that is communicated with them to find out the initial value of a node. Using this approach, secret sharing schemes such as Shamir's secret sharing[56] can be used to mask the initial value.

An eavesdropper with the capability of tapping all the channels violates the solutions presented formerly as the two approaches. In the first approach, the eavesdropper who can tap all channel simply knows all the information transmitted in the system thus, the initial value will be disclosed. In the second approach, an eavesdropper can be considered as a single adversary with the information output of all the other nodes than the targeted node. This can be considered as a graph with 2 nodes. In a graph with 2 nodes, the exact average will reveal the other one's initial value. Both cases are weak against an eavesdropper.

One way of extending the privacy to also cover the eavesdropper case is given by He et. al.[24]. They have proposed a secret continuous function that each pair of nodes agree before starting the consensus. The nodes release their first blinded state variable $x_i^+[0] = x_i[0] + \theta_i[0]$ in the same way along with a variable $z_{ij} \in \mathbb{R}$. The selection of the noise in the other iterations changes slightly. The noise for the other iterations is given by the following formula.

$$w_i(k) = \begin{cases} \gamma\theta_i[1] - \tilde{\theta}_i[0] & \text{, if } k = 1 \\ \gamma^k\theta_i(k) - \gamma^{k-1}\theta_i(k-1) & \text{, } k \geq 2 \end{cases} \tag{7.1}$$

where $\tilde{\theta}_i[0] = \theta[0] - (F_{ij}(z_{ij}) - F_{ji}(z_{ji}))$, F_{ij} represent the secret continuous function between nodes v_i and v_j. Since the sum does not change and the secret continuous functions are only known by the nodes v_i and v_j, security against an eavesdropper can be satisfied. The privacy of this algorithm has not been investigated and can be considered as future work. The problem with this approach is the distribution of the secret continuous functions. There is either a third party involved or a secure channel which can be impractical. A scalable and fully distributed way of agreeing on a secret function can extend the privacy limitation of the exact privacy-preserving consensus average algorithms. Intuitively speaking, an adversary will be able to revert the operations that any node goes through if it can capture all the transmitted messages that it receives and transmits unless there is some randomness in the way these messages are being used.

The differential privacy approach which lets the nodes reach the inexact average of the initial values, on the other

hand, can preserve the privacy of the initial values against both the semi-honest adversary and the eavesdropper. In one approach [23], it has been presented that the nodes will reach to an unbiased estimate of the initial value with an expression for the accuracy versus privacy trade-off. In systems where the exact average is not a necessity, the strongest privacy guarantee for non-cryptographic schemes is given by the differential privacy based methods.

7.5 Comparison to the Related Work

The research on solving the privacy-preserving consensus averaging problem mainly focuses on solving the problem in a synchronous fashion where at each iteration each node communicates with all its neighbors and update their state variables. Synchronous algorithms require clock synchronization and are sensitive to changes in the network topology. Instead, asynchronous algorithms do not require clock synchronization, less sensitive to changes to the network topology and have a reduced execution time per iteration. The research on privacy-preserving asynchronous algorithms is fairly new and there has been one paper[31] investigating this problem in an asynchronous setting. The authors investigate the noise-obfuscation technique and analyze the convergence properties of the algorithm without exploring the provided privacy.

Three methods have been proposed to solve the consensus averaging problem while giving a privacy guarantee to the initial values. The results are shown in section 7.5 comparing to the results in the literature. All three methods reach the exact average of the initial values. The differential privacy based methods reach an unbiased estimate of the average of the initial values. The accuracy of the estimates is traded off with the privacy guaranteed to the initial values. The differential privacy method promises security against an eavesdropper or a semi-honest adversary without any assumption on the topology or the number of honest nodes in the system. This lets the differential privacy based methods to be useful when the security against an eavesdropper is required and cryptographic methods can not be used. The third parties are costly and can be impractical in certain situations. The proposed approaches and differential

	State Decomposition	Hybrid State Decomposition	Noise Obfuscation[31]	Differential Privacy[23]	TTP[57]
Accuracy	Exact	Exact	Exact	Inexact	Inexact
Computational Complexity	Linear	Exponential	Linear	Linear	Linear
Communication Bandwidth	$O(1)$	$O(\beta)$	$O(1)$	$O(1)$	$O(1)$
Adversary Model	Passive	Passive	Passive	Passive	Passive
Assumption for Privacy	$\mathrm{N}_i \cup v_i \not\subset N_j \cup v_j$	—	$\mathrm{N}_i \cup v_i \not\subset N_j \cup v_j$	—	—
Noise Insertion	Bounded	Bounded	Unbounded	Unbounded	None
Trusted Third Party	No				Yes

Table 7.1: Table for the comparison of privacy-preserving consensus averaging algorithms

privacy approach do not require the existence of a trusted third party. In the last column of section 7.5, the authors

use a trusted third party to apply the differential privacy using a cloud-based noise insertion process. Instead of the nodes in the system blinding their state variable, they send it to the cloud where the noise insertion occurs. The cloud obfuscates the received state variables according to the differential privacy requirements before sending it to the nodes to solve the consensus problem. In this case, nodes do not sample a new noise at each iteration which would reduce the complexity of the execution on nodes' side.

The convergence rate of the state decomposition and hybrid state decomposition reduces due to the increase in the number of nodes in the network after the decomposition. However, assuming that the nodes merge their state variables after the initialization, the convergence rate can be kept the same as the non-privacy preserving version. The noise-obfuscation technique has the same convergence rate as the standard randomized gossip given that the noise decay rate satisfies the constraint given in eq. (6.2). The differential privacy based methods[23] has the same or lower convergence rate compared to the standard version where the bottleneck is the worst-case decay rate of the noise sequence among the agents. An asynchronous differentially private consensus averaging algorithm which investigates the bounds on the decay rate such that the convergence rate is not dominated by the decay rate can be investigated.

It can be seen in the section 7.5 that all the methods except the hybrid state decomposition approach have lower computational complexity and communication bandwidth. The reason is the inclusion of the homomorphic encryption to the consensus process. The hybrid state decomposition approach, however, has a bounded number of encryption operations. It can be thought as a preprocessing step which handles the privacy. After the preprocessing step, the standard algorithms that solve the consensus problem can be used. For this reason, there is a bounded amount of operations that require high computational complexity and high communication bandwidth.

Bibliography

[1] 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, Jan 2007.

[2] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184, Dec 2005.

[3] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6698–6703, Dec 2005.

[4] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 63–70, April 2005.

[5] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, March 2006.

[6] Y. Cao and W. Ren. Distributed coordinated tracking with reduced interaction via a variable structure approach. *IEEE Transactions on Automatic Control*, 57(1):33–48, Jan 2012.

[7] A. Jadbabaie, Jie Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.

[8] F. Xiao and L. Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control*, 53(8):1804–1816, Sep. 2008.

[9] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Trans. Netw.*, 14(SI):2508–2530, June 2006.

[10] Ruiliang Zhang and James T. Kwok. Asynchronous distributed admm for consensus optimization. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1701–II–1709. JMLR.org, 2014.

[11] Bingsheng He, Hong-Kun Xu, and Xiaoming Yuan. On the proximal jacobian decomposition of alm for multiple-block separable convex minimization problems and its relationship to admm. *Journal of Scientific Computing*, 66(3):1204–1217, Mar 2016.

[12] G. Zhang and R. Heusdens. Distributed optimization using the primal-dual method of multipliers. *IEEE Transactions on Signal and Information Processing over Networks*, 4(1):173–187, March 2018.

[13] J. Lin, A. S. Morse, and B. D. O. Anderson. The multi-agent rendezvous problem. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 2, pages 1508–1513 Vol.2, Dec 2003.

[14] C. Zhao, J. Chen, J. He, and P. Cheng. Privacy-preserving consensus-based energy management in smart grids. *IEEE Transactions on Signal Processing*, 66(23):6162–6176, Dec 2018.

[15] Q. Shi and C. He. Distributed source localization via projection onto the nearest local minimum. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2553–2556, March 2008.

[16] A. Alanwar, Y. Shoukry, S. Chakraborty, P. Martin, P. Tabuada, and M. Srivastava. Proloc: Resilient localization with private observers using partial homomorphic encryption. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 41–52, Los Alamitos, CA, USA, apr 2017. IEEE Computer Society.

[17] Gonzalo Mateos, Juan Andrés Bazerque, and Georgios B. Giannakis. Distributed sparse linear regression. *Trans. Sig. Proc.*, 58(10):5262–5276, October 2010.

[18] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[19] Zhenqi Huang, Sayan Mitra, and Nitin Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, ICDCN '15, pages 4:1–4:10, New York, NY, USA, 2015. ACM.

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

[21] M. Kishida. Encrypted average consensus with quantized control law. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 5850–5856, Dec 2018.

[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] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés. Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design. *Automatica*, 81:221–231, 2017.

[24] Jianping He, Lin Cai, Chengcheng Zhao, and Peng Cheng. Privacy-preserving average consensus: Privacy analysis and optimal algorithm design. *IEEE Transactions on Signal and Information Processing over Networks*, PP, 09 2016.

[25] Yongqiang Wang. Privacy-preserving average consensus via state decomposition. *IEEE Transactions on Automatic Control*, 2019.

[26] Yilin Mo and Richard M. Murray. Privacy preserving average consensus. *IEEE Trans. Automat. Contr.*, 62(2):753–765, 2017.

[27] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 784–796, New York, NY, USA, 2012. ACM.

[28] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free xor gates and applications. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, pages 486–498, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[29] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 535–548, New York, NY, USA, 2013. ACM.

[30] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy Rothblum. Differential privacy under continual observation. pages 715–724, 06 2010.

[31] Filip Hanzely, Jakub Konečnỳ, Nicolas Loizou, Peter Richtárik, and Dmitry Grishchenko. Privacy preserving randomized gossip algorithms. *arXiv preprint arXiv:1706.07636*, 2017.

[32] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems Control Letters*, 53(1):65 – 78, 2004.

[33] 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.

[34] Chunlei Zhang, Muaz Ahmad, and Yongqiang Wang. Admm based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, 14(3):565–580, 2019.

[35] Shripad Gade and Nitin H. Vaidya. Private learning on networks: Part II. *CoRR*, abs/1703.09185, 2017.

[36] M. Ruan, H. Gao, and Y. Wang. Secure and privacy-preserving consensus. *IEEE Transactions on Automatic Control*, pages 1–1, 2019.

[37] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, Oct 1949.

[38] Oded Goldreich. *Foundations of Cryptography*, volume 1. Cambridge University Press, 2001.

[39] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'99, pages 223–238, Berlin, Heidelberg, 1999. Springer-Verlag.

[40] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In Ian Goldberg and Mikhail J. Atallah, editors, *Privacy Enhancing Technologies*, pages 235–253, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[41] Rafail Ostrovsky and William E. Skeith. Private searching on streaming data. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, pages 223–240, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[42] Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. On cca-secure somewhat homomorphic encryption, 2012.

[43] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 537–554, Berlin, Heidelberg, 1999. Springer-Verlag.

[44] Keita Emura, Goichiro Hanaoka, Go Ohtake, Takahiro Matsuda, and Shota Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography – PKC 2013*, pages 32–50, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[45] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69:066138, Jun 2004.

[46] Martin Vejmelka and Milan Palu. Inferring the directionality of coupling with conditional mutual information. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 77 2 Pt 2:026214, 2008.

[47] L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Probl. Inf. Transm.*, 23(1-2):95–101, 1987.

[48] Greg Ver Steeg. Non-parametric entropy estimation toolbox. https://github.com/gregversteeg/NPEET, 2014.

[49] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Erratum: Estimating mutual information [phys. rev. e 69, 066138 (2004)]. *Phys. Rev. E*, 83:019903, Jan 2011.

[50] Martin Vejmelka and Milan Palus. Inferring the directionality of coupling with conditional mutual information. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 77:026214, 03 2008.

[51] Martin Haenggi, Jeffrey G Andrews, François Baccelli, Olivier Dousse, and Massimo Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE Journal on Selected Areas in Communications*, 27(7):1029–1046, 2009.

[52] W. J. Reichmann. *Use and abuse of statistics*. Methuen London, 1961.

[53] Leo A Goodman. On the exact variance of products. *Journal of the American statistical association*, 55(292):708–713, 1960.

[54] Hichem Kenniche and Vlady Ravelomanana. Random geometric graphs as model of wireless sensor networks. pages 103 – 107, 03 2010.

[55] Ashish Goel, Sanatan Rai, and Bhaskar Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 580–586. ACM, 2004.

[56] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[57] Matthew Hale and Magnus Egerstedt. Differentially Private Cloud-Based Multi-Agent Optimization with Constraints. *arXiv e-prints*, page arXiv:1708.08422, Aug 2017.

.1 Appendix

Privacy-Preserving Asynchronous Distributed Average Consensus via State Decomposition

Metin Calis
Circuits and Systems Group
Delft University of Technology
M.Calis@student.tudelft.nl

Richard Heusdens
Circuit and Systems Group
Delft University of Technology
R.Heusdens@tudelft.nl

Richard Hendriks
Circuit and Systems Group
Delft University of Technology
R.C.Hendriks@tudelft.nl

***Abstract*—The average consensus algorithm is used in many distributed systems such as distributed optimization, sensor fusion and the control of dynamic systems. Consensus algorithms converge through an explicit exchange of state variables. In some cases however, the state variables can be confidential. In this paper, a privacy-preserving asynchronous distributed average consensus method is proposed which decomposes the initial values into two states called the alpha and beta states such that their sum is twice the initial value. The alpha states are used to communicate with the other nodes while the beta states are used internally. Although beta states are not shared, they are used in the update of the alpha states. Unlike differential privacy based methods, the proposed algorithm achieves the exact average consensus while providing privacy to the initial values. As the variances of coupling weights go to infinity, the semi-honest adversary does not have any range to estimate the initial value of node j given that there is at least one coupling weight hidden from the adversary.**

I. Introduction

Consensus problem in dynamic systems has been a topic of interest that has found usage in many research areas allowing multiple agents to reach an agreement through local information exchange between the agent and its neighbors [1]. Some of these research areas are sensor fusion [2] [3] [4], control of swarms and flocks [5] [6], alignment problem [7] and asynchronous consensus [8]. The traditional consensus algorithms explicitly exchange their state variables to solve a common function. However, for some consensus problems such as the multi-rendezvous problem [9] or energy management in smart grids [10] the initial states can be confidential. In the former, the agents might not want to reveal their initial locations while in the latter, the energy companies might not want to reveal their individual generation rates. The challenge to solve the consensus problem while giving individual nodes a privacy guarantee initiated the new privacy-preserving distributed optimization research area.

The research directed towards solving the consensus problem while preserving the privacy of initial values can be categorized into two approaches: cryptographic [11] [12] and non-cryptographic methods [13] [14] [15] [16]. Most of the cryptographic methods use homomorphic encryption to encrypt the states that are being transferred. Due to the redundancy and randomness introduced in the ciphertext, the cryptographic algorithms provide high dimensional security at the expense of computational complexity. Another technique is to use garbled circuits [17] to implement the consensus function using a privacy-preserving manner. Although many optimization techniques [18] [19] are proposed, the garbled circuits are also computationally complex and slow compared to their non-cryptographic counterparts.

In control and real-time dynamic systems where processing time is limited, or distributed solvers which solve optimization problems iteratively, cryptographic methods are not suitable due to the time the encryption and the decryption takes. To reduce the time and complexity, privacy-preserving non-cryptographic consensus methods are proposed. The methods to solve this problem can be categorized into three parts: differential privacy [13] [20], noise-obfuscation [14] [16] and transformation methods [15]. Differential privacy based approaches trade accuracy for privacy. The nodes add noise to the transmitted states and provide a differential privacy guarantee as defined in [21] or in [22] for continuous data observations. However, as proven by [13] differential privacy and exact consensus cannot be achieved simultaneously. The noise-obfuscation methods on the other hand, add correlated noise to the transmitted states. As the added noise is zero-sum and decaying in magnitude, the exact average can be achieved. The privacy is analyzed by examining the covariance matrix of the maximum likelihood estimate [16] and extended to (ϵ, δ) privacy [14] where ϵ and δ represent the range and estimation confidence respectively. In this paper, we will show that the state decomposition method of [15] acts like a transformation method where the individual states are transformed into another domain via additive noise. In this case, privacy is defined to be the lack of any guaranteed estimation accuracy of the initial values which is also followed here.

Some networks are constrained by few characteristics which prevent an application of a synchronous algorithm. The network might lack a centralized entity which processes all the information and synchronizes the time, the power resources and compute power may be limited or the network topology can be varying. For this reason, asynchronous distributed optimization techniques such as gossip algorithms [23] and convex optimization based algorithms [24] [25] were proposed. To the best of our knowledge only [26] examines the privacy

Some parts of this paper has been presented in the Audio Analysis Workshop, 2019, Aalborg hosted by Aalborg University.

of initial values in a randomized gossip setting using noise-obfuscation techniques. The authors analyze the convergence conditions and rate of convergence without quantifying the provided privacy.

In this paper, we introduce an asynchronous privacy-preserving randomized gossip algorithm via the state decomposition technique proposed by [15]. Information-theoretic privacy analysis is done which shows that as the variances of the coupling weights go to infinity, there is no range to estimate the initial values with any guaranteed accuracy. This guarantee is given with the assumption that there is at least one node that the adversaries cannot listen to in the neighborhood of the target node. This assumption is based on the results found by [14] [16] and it is also followed in this paper. The proposed approach does not need any centralized entity or a trusted third party. Due to its asynchronous nature, computational power and energy are distributed and it is more robust to changes in the network topology. It has been assumed that there are no channel encryptions which makes the system suitable for wireless sensor networks where the channel encryption is costly and computationally complex.

II. Preliminaries and Problem Formulation

A. Average Consensus

The undirected graph G is represented as $G = (V, E)$ with nodes being represented as $V = \{v_1, v_2, ..., v_N\}$ and the edge set as $E \subset V \times V$. The ith component of the vector $x[k] = [x_1[k], x_2[k], ..., x_N[k]]$ represent the state of node i at the iteration k. The set of neighbors of node i is $N_i = \{v_j \in V : (v_i, v_j) \in E\}$ and its cardinality is shown as $|N_i|$. The goal is to compute $x_{\text{ave}} = \sum_{j=1}^{N} x_j[0]$ using an asynchronous algorithm while hiding the initial states. The privacy is analyzed against attacks by a passive adversary and an eavesdropper. Throughout the paper the following assumption holds.

Assumption 1: The graph is connected, undirected and there are no channel encryptions in the network.

One way to solve the average consensus in an asynchronous fashion is the randomized gossip algorithm [23] with the iteration

$$x_i[k+1] = x_i[k] + \frac{1}{2}(x_j[k] - x_i[k]). \tag{1}$$

Under Assumption 1, it is proven that the state variables converge to $x_{\text{ave}} = \sum_{j=1}^{N} x_j[0]$ in finite time.

B. Adverserial Model

Semi-honest adversary or passive adversary is defined to be a node in the network who follows the protocol steps correctly but tries to gain more information by collecting the data they receive. The information that semi-honest adversary knows is its own internal state variables and the broadcasted messages within its neighborhood.

An eavesdropper is defined to be an adversary who is able to tap all communication channels. The eavesdropper knows all the shared data however it does not know the internal state variables that are not shared in the system.

III. Asynchronous State Decomposition Approach

Each node decomposes its state value, say $x_i[0] \in \mathbb{R}$ into two substates $x_i^{\alpha}[0] \in \mathbb{R}$ and $x_i^{\beta}[0] = 2x_i[0] - x_i^{\alpha}[0]$ resulting in an increase in the number of nodes from N to $2N$. $x^{\alpha}[k]$ is used in the interaction with the other nodes, while $x^{\beta}[k]$ is used as an internal update. Although $x^{\beta}[k]$ is never shared, it is used in the evaluation of $x^{\alpha}[k]$. Using the state decomposition approach [15], the randomized gossip update (1) becomes

$$\begin{aligned} x_i^{\alpha}[k+1] &= x_i^{\alpha}[k] + \frac{1}{3}(x_j^{\alpha}[k] - x_i^{\alpha}[k]) \\ &\quad + \frac{1}{3}(x_i^{\beta}[k] - x_i^{\alpha}[k]), \\ x_i^{\beta}[k+1] &= x_i^{\beta}[k] + \frac{1}{3}(x_i^{\alpha}[k] - x_i^{\beta}[k]) \end{aligned} \tag{2}$$

subject to $x_i^{\alpha}[0] + x_i^{\beta}[0] = 2x_i[0]$.

Forcing the update weights to be between $(0, 1)$ limits the privacy that can be provided. For this reason, two phases are introduced: the initialization phase and the consensus phase. In the initialization phase, the update weights are selected from the set of all real numbers with the condition that the sum of all state variables never changes. Selecting the coupling weights from the set of all real numbers introduce randomness to the system that will provide the privacy of the initial values. The convergence rate does not get affected but the initial errors get bigger. As the sum of the state variables does not change, the exact consensus can still be achieved.

At consensus phase the update equations are the same as (2). As privacy is already established in the initialization phase, the motivation is to let nodes reach to the average of their state values in finite time. When v_i goes through the initialization update once with all its neighbors, it proceeds to the consensus phase.

A. Initialization Phase

During the initialization phase, the coupling weights are selected from the set of all real numbers. The update equations become

$$\begin{aligned} x_i^{\alpha}[k+1] &= x_i^{\alpha}[k] + a_{ij}[k](x_j^{\alpha}[k] - x_i^{\alpha}[k]) \\ &\quad + a_{i,\alpha\beta}[k](x_i^{\beta}[k] - x_i^{\alpha}[k]), \\ x_i^{\beta}[k+1] &= x_i^{\beta}[k] + a_{i,\alpha\beta}[k](x_i^{\alpha}[k] - x_i^{\beta}[k]) \end{aligned} \tag{3}$$

where $a_{i,\alpha\beta}[k] \in \mathbb{R}$ and $a_{ij}[k] \in \mathbb{R}$. The node v_i that will update its state variable is selected with equal probability $p_i = 1/N$. The node v_i selects a neighboring node v_j with probability $p_{j|i} = \frac{1}{|N_i \setminus S_i|}$ where N_i is the set of neighbors of v_i, S_i is the set of neighbors of v_i that it has gone through initialization update and $\setminus$ is the set difference operator. The selected nodes update their alpha and beta states using (3) at iteration k while all the other nodes keep their states the same. The initialization phase of v_i ends only when it has gone through the update (3) once with all $v_j \in N_i$. If there are no neighbors left to go through the initialization, v_i selects from the set of neighbors that also finished initialization to start the consensus phase.

Remark 1: The coupling weights are symmetric at each iteration, i.e. $a_{ij}[k] = a_{ji}[k]$. To achieve this equality, v_i picks $a_{i\to j} \in \mathbb{R}$ randomly and sends it to v_j. The multiplication $a_{ij} = a_{i\to j}a_{j\to i}$ is the shared coupling weight that will be used by v_i and v_j.

B. Consensus Phase

During the consensus phase, the nodes update their state variables to reach to the average of their initial values. The node v_i is selected with equal probability $p_i = 1/N$. If v_i has gone through the initialization update with all its neighbors once, it selects a neighboring node v_j with equal probability $p_{j|i} = \frac{1}{|F_i|}$ where F_i is defined to be the set of neighbors of v_i that have finished initialization. Given that F_i is not empty, nodes v_i and v_j go through the consensus update defined in (2). If there is no neighbor that has finished initialization, the update is skipped.

Remark 2: Initialization is finished when a node updates its state value using (3) once with all its neighbors. This is done to make sure that the node that the adversary cannot listen to is communicated.

Remark 3: The coupling weights are fixed to $1/3$ which means that after the initialization, the beta state values will be disclosed. However as shown in Section IV, privacy has already been established during the initialization phase.

Theorem 1: Under Algorithm 1 and Assumption 1, the proposed algorithm converges to the exact average of the initial values.

Proof. Since the coupling weights are symmetric, the sum of the network does not change at each iteration. It can be shown that for each update during initialization

$$\frac{1}{2N}\sum_{j=1}^{N}(x_j^{\alpha}[k] + x_j^{\beta}[k]) = \frac{1}{2N}\sum_{j=1}^{N}(x_j^{\alpha}[k+1] + x_j^{\beta}[k+1]).$$

After the initialization, the convergence analysis is based on [27] and [15]. Four requirements are given in [27] for the exact consensus to be achieved. These properties are shown to hold as follows.

1) *Weight Rule:* There exists a scalar η with $0 < \eta < 1$ such that for every iteration after the initialization, all nonzero $a_{ij}[k]$ satisfy $\eta \leq a_{ij}[k] < 1$ and all nonzero $a_{i,\alpha\beta}[k]$ satisfy $\eta \leq a_{i,\alpha\beta}[k] < 1$. In fact, both $a_{ij}[k]$ and $a_{i,\alpha\beta}[k]$ are fixed to $1/3$ which is between $(0, 1)$. The rest of the coupling weights are set to zero.
2) *Doubly Stochasticity:* For every update, three out of all the coupling weights are $1/3$ where the rest is zero. Since all the coupling weights are also symmetric, the sum is preserved and $(1, \mathbf{1})$ is an eigenpair of the weight matrix.
3) *Connectivity:* The graph is connected before state decomposition because of the Assumption 1. State decomposition creates a connected graph since each node decomposes itself into two substates which are also connected. Thus, the graph is connected.
4) *Bounded Intercommunication Interval:* If Algorithm 1 is followed, the nodes go through the initialization unless they have completed initialization with all its neighbors. As the selection of nodes are with equal probability and they select neighbor nodes that they haven't connected, it is expected that all nodes will finish initialization in finite iterations. During the consensus phase, the nodes go through the update with the neighbors that they know who has finished initialization. Since the selection of nodes is at random, in a bounded time B, each node is expected to be contacted at least once.

As four of the requirements hold, all substates will converge to the mean $\frac{1}{2N}\sum_{j=1}^{N}(x_j^{\alpha}[k] + x_j^{\beta}[k])$ which is equivalent to $\frac{1}{N}\sum_{j=1} x_j[0]$ due to the initial constraint $x_i^{\alpha}[0] + x_i^{\beta}[0] = 2x_i[0]$.

$$\lim_{k\to\infty} x_i^{\alpha}[k] = \lim_{k\to\infty} x_i^{\beta}[k] = \frac{1}{N}\sum_{j=1} x_j[0]$$

Algorithm 1 Asynchronous State Decomposition Consensus

Let S_i to define the set of neighbors that v_i has selected before.
Let F_i be the set of neighbors of v_i that have finished initialization.

The initial values are decomposed into $x[0] \in \mathrm{R}$ and $x^{\beta}[0] = 2x[0] - x^{\alpha}[0]$
for k=1,...,K **do**
  Select v_i with probability $p_i = \frac{1}{N}$
  if $S_i \neq N_i$ **then**
    Select $v_j \in N_i \setminus S_i$ with $p_{j|i} = \frac{1}{|N_i \setminus S_i|}$
    Add v_j to S_i
    Select $a_{i\to j} \in R$ and broadcast $x_i^{\alpha}[k], a_{i\to j}$
    Calculate $a_{ij} = a_{i\to j}a_{j\to i}$
    Select $a_{i,\alpha\beta} \in R$
    Update using (3)
    Broadcast if $S_i = N_i$
  else if $F_i \neq \varnothing$ **then**
    Select $v_j \in F_i$ with probability $p_{j|i} = \frac{1}{|F_i|}$
    Broadcast $x_i^{\alpha}[k]$
    Update using (2)
  else
    Skip the iteration
  end if
end for

IV. Privacy Analysis

Following the convention in [15], the privacy is defined as the following.

***Definition** 1:* The privacy of the initial value $x_i[0]$ is preserved if an adversary cannot estimate the value of $x_i[0]$ with any guaranteed accuracy.

A. Privacy Against Semi-Honest Adversary

The privacy breach is explained by [16] which shows that if all the neighbors of v_i can be listened by the passive adversary, the privacy cannot be established. The following topological

assumption is made to prove privacy however, it is shown that it only needs to hold during the initialization phase.

Assumption 2: For all $v_i \neq v_j$, there is at least one neighbor of v_i that is not the neighbor of v_j.

$$N_i \cup v_i \not\subset N_j \cup v_j$$

Theorem 2: Under Algorithm 1, Assumptions 1 and 2, the privacy as defined in Definition 1 will be achieved asymptotically as the variances of coupling weights go to infinity.

Proof. Let $x_i[0]$ be the initial value that the adversary tries to estimate. The initial value can be found using the relation $2x_i[0] = x_i^\alpha[0] + x_i^\beta[0]$. Since $x_i^\alpha[0]$ is released and known by the adversary, estimating $x_i[0]$ is the same as estimating $x_i^\beta[0]$. There are two cases which defines the privacy of the initial value. First one is the updates happening within the neighborhood of the adversary and the second one is the update happening outside the neighborhood of the adversary.

If the selected nodes are inside the neighborhood of the adversary, $x_i^\alpha[k]$, $x_j^\alpha[k]$ and $a_{ij}[k]$ will be known by the adversary and can be treated as a constant. The information leakage for this case can be defined as the following.

$$I(x_i^\alpha[k+1]; x_i^\beta[k] | x_i^\alpha[k], a_{ij}[k], x_j^\alpha[k])$$

Using (3), the conditional mutual information becomes

$$I(a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k]); x_i^\beta[k] | x_i^\alpha[k])\,.$$

There should be no information leakage regarding the $x_i^\beta[k]$ during this case since $x_i^\beta[k]$ is directly related to $x_i[0]$ due to the initial constraint. To establish the privacy, the following should be shown.

$$\lim_{\sigma^2_{a_{i,\alpha\beta}[k]} \to \infty} I(a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k]); x_i^\beta[k] | x_i^\alpha[k]) = 0 \quad (4)$$

The second case is when v_i contacts v_m that cannot be listened by the adversary. Let $s[k]$ denote the sum $\sum_{k=1}^{T} a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k])$ that can be obtained by

$$s[k+1] = s[k] + x_i^\alpha[k+1] - a_{ij}[k](x_j^\alpha[k] - x_i^\alpha[k]) \quad (5)$$

where $s[0] = x_i^\alpha[0]$ and T is the iteration that v_i has finished the initialization phase. The initial value can be found by the following relation

$$\hat{x}_i[0] = \frac{1}{2}(s[T] + x_i^\beta[T]) \quad (6)$$

as the first consensus update after the initialization discloses $x_i^\beta[T]$, due to the fixed coupling weights. The adversary cannot go through the update (5) in this case, since only $x_i^\alpha[k]$ is known. The Assumption 2 lets $a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k])$ to be blinded by $a_{ij}[k](x_j^\alpha[k] - x_i^\alpha[k])$ at least once. For this case the information leakage can be defined as the following.

$$I(x_i^\alpha[k+1]; a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k]) | x_i^\alpha[k])$$

Using (3), the conditional mutual information becomes

$$I(a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k]) + a_{ij}[k](x_j^\alpha[k] - x_i^\alpha[k]); a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k]) | x_i^\alpha[k]).$$

For the privacy to be established, the following should be shown.

$$\lim_{\substack{\sigma^2_{a_{ij}[k]} \to \infty \\ \sigma^2_{a_{i,\alpha\beta}[k]} \to \infty}} I(a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k]) + a_{ij}[k](x_j^\alpha[k] - x_i^\alpha[k]); a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k]) | x_i^\alpha[k]) \quad (7)$$

If (4) and (7) is shown to hold, there will be no information leakage in the system. The mutual information $I(x_i^\alpha[T]; x_i[0])$ will be zero since there will be no dependence between the alpha states and the initial value.

For the simplicity of notation let $W_{\alpha\beta}$ denote $a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k])$ and let W_{ij} denote $a_{ij}[k](x_j^\alpha[k] - x_i^\alpha[k])$. The iteration number $[k]$ is omitted in the equations and only written to explicitly state the next iteration or the iteration 0.

First, it will be shown that for a fixed bounded variance $x_i^\alpha[0]$, the conditional mutual information $I(x_i^\alpha[k+1]); x_i^\beta[k] | x_i^\alpha[k])$ goes to zero as the variance of $a_{i,\alpha\beta}[k]$ goes to infinity. Let $x_i^\alpha[0]$ be a continuous random variable with $\sigma^2_{x_i^\alpha[0]} < \infty$. Define $\gamma = \frac{1}{\sigma^2_{a_{i,\alpha\beta}[k]}}$ and $\bar{W}_{\alpha\beta} = \gamma a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k])$. The conditional mutual information can be written as the following.

$$I(x_i^\alpha[k+1]; x_i^\beta | x_i^\alpha, a_{ij}, x_j^\alpha) = I(W_{\alpha\beta}; x_i^\beta | x_i^\alpha)$$

The mutual information is invariant to scaling.

$$I(\gamma W_{\alpha\beta}; \gamma x_i^\beta | \gamma x_i^\alpha) = I(\bar{W}_{\alpha\beta}; \gamma x_i^\beta | \gamma x_i^\alpha)$$

As the variance of $a_{i,\alpha\beta}[k]$ goes to infinity, the conditional mutual information will go to zero.

$$\begin{aligned}\lim_{\sigma^2_{a_{i,\alpha\beta}} \to \infty} I(\bar{W}_{\alpha\beta}; \gamma x_i^\beta | \gamma x_i^\alpha) &= \lim_{\gamma \to 0} I(\bar{W}_{\alpha\beta}; \gamma x_i^\beta | \gamma x_i^\alpha) \\ &= I(\bar{W}_{\alpha\beta}; 0) = 0\end{aligned}$$

The second case is the update happening outside the neighborhood of the adversary. Define $\beta = \frac{1}{\sigma^2_{a_{ij}[k]}}$ and $\bar{W}_{ij} = \beta a_{ij}[k](x_i^\beta[k] - x_i^\alpha[k])$. Mutual information is invariant to scaling.

$$\begin{aligned}I(x_i^\alpha[k+1]; W_{\alpha\beta} | x_i^\alpha) &= I(\gamma\beta x_i^\alpha[k+1]; \gamma\beta W_{\alpha\beta} | \gamma\beta x_i^\alpha) \\ &= I(\beta\bar{W}_{\alpha\beta} + \gamma\bar{W}_{ij}; \beta\bar{W}_{\alpha\beta} | \gamma\beta x_i^\alpha)\end{aligned}$$

When the variances of both coupling weights go to infinity, the conditional mutual information will go to zero.

$$\lim_{\substack{\sigma^2_{a_{ij}} \to \infty \\ \sigma^2_{a_{i,\alpha\beta}} \to \infty}} I(\beta\bar{W}_{\alpha\beta} + \gamma\bar{W}_{ij}; \beta\bar{W}_{\alpha\beta} | \gamma\beta x_i^\alpha) =$$

$$\lim_{\substack{\gamma \to 0 \\ \beta \to 0}} I(\beta\bar{W}_{\alpha\beta} + \gamma\bar{W}_{ij}; \beta\bar{W}_{\alpha\beta} | \gamma\beta x_i^\alpha) = I(0;0) = 0$$

Let T be the iteration at which the initialization has ended. $\mathbf{x}^\alpha[T]$ represents the vector of alpha values obtained starting from $\mathbf{x}[0]$. During the consensus phase let W^k denote the information obtained at each iteration to deduce $\mathbf{x}[0]$ with $k =$

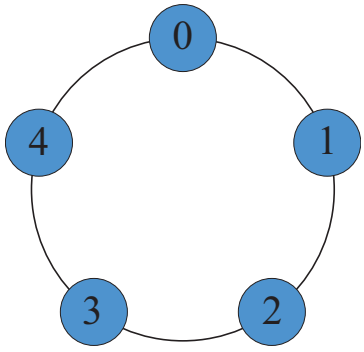

Fig. 1: Network Topology.

$\{1, 2, ...K\}$ where K is the total iteration number. The final mutual information can be represented as $I(\mathbf{x}[0]; W^k)$. Fixing the update weights enables to find a function $F^k(\mathbf{x}^\alpha[T]) = W^k$ that will take the $\mathbf{x}^\alpha[T]$ as input and will create the output W^k. The random variables will create a Markov chain $\mathbf{x}[0] \rightarrow \mathbf{x}^\alpha[T] \rightarrow W^k$ for $k = \{1, 2, ...K\}$. Thus data processing inequality suggests that

$$I(\mathbf{x}[0]; \mathbf{x}^\alpha[T]) \geq I(\mathbf{x}[0]; W^k), \quad k = 1, ..., K.$$

$I(\mathbf{x}[0]; \mathbf{x}^\alpha[T])$ is shown to be going to zero earlier. Any clever manipulation of data cannot increase the mutual information. Thus, given that there is at least one neighbor of node i that is not in the neighborhood of the adversary, the semi-honest adversary cannot estimate the initial value of node i with any guaranteed accuracy. In fact, the only requirement is that one of the coupling weight $a_{ij}[k]$ is hidden from the adversary during the initialization phase.

B. Privacy Against Eavesdropper

The eavesdropper is defined to be an adversary who taps all the communications in the channel. Following the results of [16], the eavesdropper will be able to deduce the initial value of v_i exactly. The eavesdropper will get $\sum_{k=1}^{T} a_{i,\alpha\beta}[k](x_i^\beta[k] - x_i^\alpha[k])$ via the update (5) and find the initial value using (6).

Remark 4: It has been shown at Section IV that if one of the $a_{ij}[k]$ is hidden during the initialization phase, the privacy can be achieved. This can be done using a one-time encryption during the initialization process. An asynchronous algorithm that would provide protection against eavesdropper with the minimum amount of encryption is left for future work.

V. Experiments

To demonstrate the performance of the proposed approach, 5 node cyclic graph as shown in Fig 1 is selected. The semi-honest adversary won't be able to deduce the initial values exactly however, the privacy as defined in Definition 1 will not hold as the variances of coupling weights are bounded. The $x^\alpha[0]$ and the coupling weights $a_{i,\alpha\beta}[k]$, $a_{ij}[k]$ are selected from $[-5, 5]$ while the initial values are selected from $[0, 20]$ randomly. Fig 2 shows the final estimate of v_2 guessing the initial value of v_1. The adversary goes through the update of (5) while assigning 0 to $a_{1,\alpha\beta}(x_1^\beta[k] - x_1^\alpha[k])$ when v_1 contacts v_0.

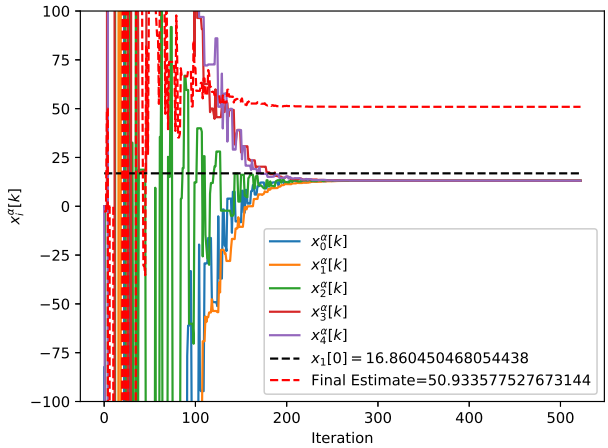

Fig. 2: Passive adversary estimation performance plot. The flat lines represent the convergence process. The horizontal dashed line represent the initial value of v_1 which the adversary tries to estimate.

Remark 5: The least amount of privacy is guaranteed when the first initialization update is between the target node and its neighbor that cannot be listened by the adversary. The variance of alpha state values increase as the iterations continue resulting in an increase in the variance of $a_{ij}[k](x_j^\alpha[k] - x_i^\alpha[k])$. Although no formal definition of privacy is given, it is expected that the estimates will be worse if the node that cannot be listened is contacted later in the initialization.

As seen in Fig 3, the convergence rate of the proposed approach is lower than the standard randomized gossip without privacy-preserving attribute because of the increase in the number of nodes due to the state decomposition. Given that each node merges their state variables into one by $x_i[k] = (x_i^\alpha[k] + x_i^\beta[k])/2$, the same convergence rate to the standard randomized gossip can be obtained. Differential privacy and noise-obfuscation methods promise the same convergence rate with an offset due to the initial errors introduced to the system.

The proposed approach promises convergence to the exact average, unlike differential privacy based methods. In addition to this, the privacy is guaranteed given that one of the coupling weight $a_{ij}[k]$ is hidden from the adversary during the initialization process. Unlike the noise-obfuscation based methods, the Assumption 2 needs to hold only during the initialization phase.

VI. Conclusions

In this paper, an asynchronous privacy-preserving average consensus algorithm is proposed using the state decomposition approach. An information theoretic privacy analysis is done which promises to preserve the privacy of initial values given that there is at least one coupling weight hidden from the adversary. The proposed approach converges to the exact average however, the convergence rate drops due to the increase in the number of nodes. Future research includes relaxing

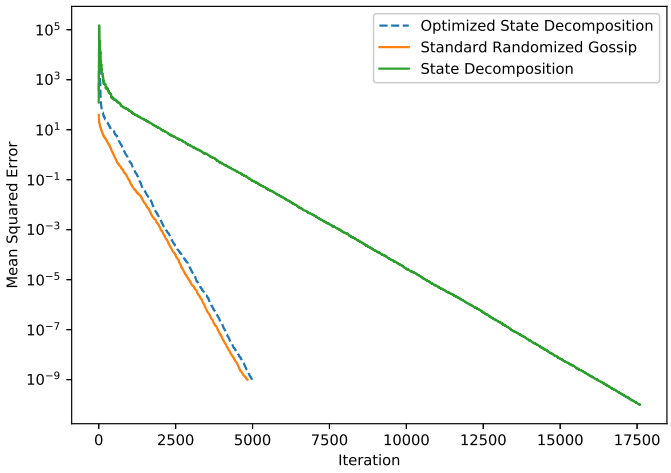

Fig. 3: Convergence rate plot for the proposed approach and the standard randomized gossip.

the topological assumption and providing a privacy-preserving scheme robust against eavesdropping attacks.

REFERENCES

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[2] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 8179–8184.

[3] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 6698–6703.

[4] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, April 2005, pp. 63–70.

[5] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

[6] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 33–48, Jan 2012.

[7] A. Jadbabaie, Jie Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[8] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, Sep. 2008.

[9] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, Dec 2003, vol. 2, pp. 1508–1513 Vol.2.

[10] C. Zhao, J. Chen, J. He, and P. Cheng, "Privacy-preserving consensus-based energy management in smart grids," *IEEE Transactions on Signal Processing*, vol. 66, no. 23, pp. 6162–6176, Dec 2018.

[11] M. Kishida, "Encrypted average consensus with quantized control law," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 5850–5856.

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

[13] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés, "Differentially private average consensus: Obstructions, trade-offs, and optimal algorithm design," *Automatica*, vol. 81, pp. 221–231, 2017.

[14] Jianping He, Lin Cai, Chengcheng Zhao, and Peng Cheng, "Privacy-preserving average consensus: Privacy analysis and optimal algorithm design," *IEEE Transactions on Signal and Information Processing over Networks*, vol. PP, 09 2016.

[15] Yongqiang Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Transactions on Automatic Control*, 2019.

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

[17] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway, "Foundations of garbled circuits," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, New York, NY, USA, 2012, CCS '12, pp. 784–796, ACM.

[18] Vladimir Kolesnikov and Thomas Schneider, "Improved garbled circuit: Free xor gates and applications," in *Automata, Languages and Programming*, Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, Eds., Berlin, Heidelberg, 2008, pp. 486–498, Springer Berlin Heidelberg.

[19] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner, "More efficient oblivious transfer and extensions for faster secure computation," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, New York, NY, USA, 2013, CCS '13, pp. 535–548, ACM.

[20] Zonghao Huang, Rui Hu, Eric Chan-Tin, and Yanmin Gong, "DP-ADMM: admm-based distributed learning with differential privacy," *CoRR*, vol. abs/1808.10101, 2018.

[21] Cynthia Dwork, "Differential privacy," in *Automata, Languages and Programming*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, Eds., Berlin, Heidelberg, 2006, pp. 1–12, Springer Berlin Heidelberg.

[22] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy Rothblum, "Differential privacy under continual observation," 06 2010, pp. 715–724.

[23] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2508–2530, June 2006.

[24] Ruiliang Zhang and James T. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. 2014, ICML'14, pp. II–1701–II–1709, JMLR.org.

[25] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 173–187, March 2018.

[26] Filip Hanzely, Jakub Konečnỳ, Nicolas Loizou, Peter Richtárik, and Dmitry Grishchenko, "Privacy preserving randomized gossip algorithms," *arXiv preprint arXiv:1706.07636*, 2017.

[27] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, April 2010.