Proceedings of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Secure Consensus against Message Manipulation Attacks in Synchronous Networks ⋆

Chengcheng Zhao * **Jianping He** * **Peng Cheng** * **Jiming Chen** *

* *State Key Lab. of Industrial Control Technology, Zhejiang University,*
Hangzhou 310027, P.R. China. Emails: zccsq90@gmail.com
{jphe, pcheng, jmchen}@iipc.zju.edu.cn

Abstract: The security problem of consensus has been attracting increasing research attention for a large number of application scenarios. Many existing solutions in this aspect rely on the assumption that most of the neighboring nodes of each safe node are honest which may not be able to satisfy various practical situations. Motivated by a recent work dealing with the secure consensus for time synchronization in asynchronous networks, this paper aims to handle secure synchronous consensus under message manipulation attacks. Specifically, we first propose a secure synchronous consensus algorithm (SSCA), and prove that SSCA converges with an exponential rate through matrix analysis. Furthermore, we investigate how the exact behavior of message manipulation attack will affect the convergence of SSCA. Specifically, we show how such result can be employed to classify the attack behavior and help analyze the consensus performance. Examples and extensive simulations are provided to evaluate the effectiveness of proposed results.

Keywords: secure synchronous consensus, verification mechanism, convergence, valid/invalid attack.

1. INTRODUCTION

Consensus has found a large number of applications, including clock synchronization (He et al. [2014a,b]), sensor information fusion (Xiao et al. [2005], Cao et al. [2008]), distributed estimation and detection (Pasqualetti et al. [2010]), and etc. In the past decade, extensive efforts have also been devoted to the related theories and algorithms (Ren et al. [2005], Zhou et al. [2013]). It should be noted that most existing works are based on the assumption that the system is deployed in benign environment. However, in harsh surroundings with various malicious attacks, such as denial-of-service and message manipulation attack (He et al. [2013]), they may become vulnerable or even invalid.

Recently, secure consensus design has attracted increasing research attention. Fabio Pasqualetti *et al.* in (Pasqualetti et al. [2012, 2010], N. H. Vaidya and Liang [2012], LeBlanc et al. [2013]) provide Intrusion Detection System (IDS) by detecting behavior of each node and isolating misbehaving nodes in time. LeBlanc and Koutsoukos utilize the information provided by the limited number of neighbors to ensure the security of consensus (LeBlanc and Koutsoukos [2011]). Zhang *et al.* propose Median Consensus Algorithm (MCA) where each node only uses the median of states from all neighbors (Zhang and Sundaram [2012]). Note that all the results may become vulnerable or even invalid when the network is weakly connected. In order to deal with this issue, Yan *et al.* propose a distributed hash-based verification mechanism for ensuring trust-worthy state update of each node. However, it still requires the assumption that the majority of nodes in the neighborhood are honest. Such assumption is relaxed in (He et al. [2013]), which develops a Secure Average-consensus-based Time Synchronization protocol (SATS) for time synchronization in asynchronous sensor networks by exploiting the two-hop information. Yet, two important questions are not answered in (He et al. [2013]), namely, 1) whether and how SATS can be extended to deal with synchronous networks; 2) how exactly the attack behavior will affect the convergence of consensus.

Motivated by the questions arising from existing works, in this paper we develop a defence mechanism inspired by the safeguard method proposed in (He et al. [2013]) in order to provide secure consensus in synchronous networks.

Furthermore, we aim to reveal the answers for why and when the message manipulation attack can accelerate the convergence of consensus. The main contributions of our paper are as follows.

1. For synchronous networks, we propose a verification process and update rules based on two-hop neighbor nodes' information, which guarantees the safe nodes always use credible information to update their states.
2. We depict the dynamics of nodes' states (including safe and attack nodes) under SSCA in matrix and prove the convergence as well as the exponential convergence rate of SSCA.
3. We analyze how attack affects the convergence of consensus under SSCA. Attack is classified into valid and invalid attack to investigate the variation of the max-min state deviation of all safe nodes for one iteration update.

The remainder of this paper is organized as follows. Section 2 provides the models and formulations of the problem, and the detailed secure synchronous consensus algorithm is presented in Section 3. Section 4 analyzes performance of SSCA. Section 5 tests main results through numerical examples and simulations. Conclusion is given in Section 6.

⋆ The work was supported in part by NSFC under Grant 61222305, the 863 High-Tech project under Grant 2011AA040101-1, the SRFDP under Grant 20120101110139, and National Program for Special Support of Top-Notch Young Professionals.

2. PRELIMINARIES AND PROBLEM FORMULATION

Consider a network with n safe nodes and m attack nodes, their communication topology is described by an undirected connected graph $G = \{V, E\}$, where $V = \{1, 2, ..., N\}$ $(n+m = N)$ is a set of N nodes and $E \subset V \times V$ is an edge set. The neighbor set of node i is denoted by $N_i = \{j|(i, j) \in E, i, j \in V\}$, i.e., $(i, j) \in E$ if and only if node j is the neighbor of node i. Let $V_s = \{1, 2, ..., n\}$ and $V_a = \{n + 1, n + 2, ..., n + m\}$ represent safe nodes set and attack nodes set, respectively. Clearly, we have $V = V_s \bigcup V_a$. Let E_s denote the edge set, where each edge in the set connects two safe nodes. An undirected graph $G_s = \{V_s, E_s\}$ consists of all safe nodes and the edges connecting between them. Assume that G_s is a static and strongly connected graph. Assume that there are more than two safe nodes in the network, i.e., $n > 2$ and m should have a lower bound, which means more details about the value of m should be given such as $n > m >= 0$. Each node is assumed to have only one corresponding ID number.

2.1 Synchronous Consensus

Different from asynchronous consensus, each node broadcasts and updates its state simultaneously in synchronous consensus, which needs to be modeled and handled in a distinct way. We give the related description as follows.

Let $X(k) = [x_1(k), x_2(k), ..., x_n(k), x_{n+1}(k), ..., x_{n+m}(k)]^T$ be the state vector of all sensor nodes, where k is the iteration times of a consensus algorithm and $X(0)$ represents the initial states of all nodes. At iteration k, let $X^s(k) = [x_1(k), ..., x_n(k)]^T$ and $X^a(k) = [x_{n+1}(k), ..., x_{n+m}(k)]^T$ be the state vector of safe nodes and attack nodes,respectively. In this paper, we consider the general consensus algorithm as follows

$$X^s(k + 1) = AX^s(k), \qquad (1)$$

where A is a row stochastic matrix, i.e., $\sum_{j=1}^{n} a_{ij} = 1, i \in V_s$. Assume $a_{ij} \in \{0\} \cup [\alpha, 1]$ $\forall i, j \in V_s$, where $a_{ii} \geq \alpha$ and $a_{ij} \geq \alpha$ when $(i, j) \in E_s$. It follows from (Olfati-Saber et al. [2007]) that under (1) all nodes' states will approximately converge to a same constant state with an exponential rate. At iteration k, while there are safe nodes using attack nodes' states for update, the minimum and maximum state of safe nodes are denoted by $h'(k) = \min[X^s(k)]$, $H'(k) = \max[X^s(k)]$. Otherwise, they are denoted by $h(k) = \min[X^s(k)]$ and $H(k) = \max[X^s(k)]$. Meanwhile, we denote the maximum deviation of all safe nodes' states for different cases as $D(k) = H(k) - h(k)$ and $D'(k) = H'(k) - h'(k)$, respectively.

2.2 Problem Setup

Most existing works concerning consensus are established under the assumption that the network is secure (all nodes are safe nodes). In reality, message manipulation attack may show up in the network, where the attack node may be a new entrant to the network or a compromised node which was a safe node before. The state of attack node is selected arbitrarily and we mainly consider two kinds of message manipulation attack, i.e., constant injection attack (CIA) and random injection attack (RIA) (He et al. [2013]). Without defence mechanism, safe nodes cannot distinguish the state of safe node and that of attack node so that they may use attack node's information to update their own states. Therefore, under the consensus algorithm (1), the form of iteration rule for each safe node can be indicated as:

$$x_i(k + 1) = \sum_{j=1}^{n} a'_{ij}(k)x_j(k) + \sum_{j=n+1}^{n+m} a'_{ij}(k)x_j(k), \forall i \in V_s, \qquad (2)$$

where each $a'_{ij}(k)$ satisfies $a'_{ij}(k) \geq \alpha$ when $(i, j) \in E$ and $a'_{ij}(k) = 0$ when $(i, j) \notin E$, and $\sum_{j=1}^{n+m} a'_{ij}(k) = 1$. Just as proven in (Khanafer et al. [2012]), any form of message manipulation attack can easily make algorithm (1) invalid, i.e., the consensus cannot be achieved by all safe nodes.

This paper considers two assumptions for attack: attack nodes only have safe neighbor nodes, i.e., there are not any two attack nodes neighboring with each other, thus no cooperation attack; attack nodes can read the information of safe nodes but cannot modify it. The second assumption can be realized by information authentication (Lu et al. [2012]).

We find that synchronous consensus cannot be solved by the method in (He et al. [2013]) for some cases. For example, when there are only two states $Q1, Q2$ in the network and each node with state $Q1$ ($Q2$) has two neighbor nodes with state $Q2$ ($Q1$), under that method, nodes with state $Q1$ will just change their states into $Q2$ while nodes with state $Q2$ will just change their states into $Q1$ according to the update rule in (He et al. [2013]). As a result, each node will always have different state from its neighbors which means that consensus cannot be achieved. Therefore, this method cannot be used for synchronous consensus directly. Hence, we need to modify the method to avoid the infinite loop of states and guarantee that synchronous consensus can be achieved by any initial states.

The first goal of this paper is to develop a secure consensus algorithm which is not constrained by the connectivity of network and guarantees that consensus can still be achieved by all the safe nodes under attack in synchronous networks. More importantly, we will investigate the fundamental problem that how the states selected by attack nodes affect the convergence of SSCA.

3. SECURE SYNCHRONOUS CONSENSUS ALGORITHM

In this section, we first introduce the verification mechanism, and then propose a secure synchronous consensus algorithm with this mechanism.

Let $\Theta_j(k), k > 0$ denote the information set of safe node j at iteration k,

$\Theta_j(k) = \{x_j(k), x_{j_m}(k-1), x_{j_M}(k-1), j, j_m, j_M\}, j \in V_s, j_m, j_M \in N_j$

where $x_{j_m}(k - 1)$ and $x_{j_M}(k - 1)$ denote the smallest and largest state among all its neighbor nodes that node j receives from its neighbor node j_m and j_M at iteration $k - 1$, respectively.

Verification Process: suppose that safe node i receives information set $\Theta_j(k), j \in N_i$. Then node i will check the following two conditions,

c_1. $j \neq j_m \neq j_M$.
c_2. $x_{j_m}(k - 1) \leq x_j(k) \leq x_{j_M}(k - 1)$.

If conditions c_1 and c_2 hold, the state $x_j(k)$ is credible for node i. Where condition c_1 ensures that nodes j_m and j_M are two different neighbor nodes of node j, and condition c_2 guarantees that node j's state is bounded by two neighbor nodes' states.

Unlike SATS (He et al. [2013]), in step 2 and step 4, we improve the update rule for the node with maximum or minimum state

Algorithm 1 : *Secure Synchronous Consensus Algorithm (SSCA)*

```
Initialization
 1. Each safe node $i$ initializes its state $x_i(0), i \in V_s$ and broadcasts it.
 2. After receiving all neighbor nodes' states, node $i$ gets the information set
    $\{x_{i_m}(0), x_{i_M}(0), i_m, i_M\}$ and updates its state as follows
     (a) If $x_i(0) < x_{i_m}(0)$, $x_i(0) = \frac{x_i(0)+x_{i_m}(0)}{2}$;
     (b) If $x_i(0) > x_{i_M}(0)$, $x_i(0) = x_{i_M}(0)$.
    Then, node $i$ broadcasts its initial information set $\Theta_i(0)$.
Iteration
 3. Node $i$ broadcasts $\Theta_i(k)$ to its neighbor nodes.
 4. Based on neighbors' states, node $i$ gets the information set
    $\{x_{i_m}(k), x_{i_M}(k), i_m, i_M\}$ and updates its state as follows
     (a) If $x_i(k) < x_{i_m}(k)$, $x_i(k) = \frac{x_i(k)+x_{i_m}(k)}{2}$;
     (b) If $x_i(k) > x_{i_M}(k)$, $x_i(k) = x_{i_M}(k)$.
 5. Node $i$ executes the verification process for its neighbors' information set
    $\Theta_j(k), j \in N_i$. If and only if information set $\Theta_j(k)$ passes the verification
    process, $x_j(k), j \in N_i$ is credible enough for node $i$ and gets stored.
 6. Node $i$ updates its state according to (2), where only a positive weight
    $a'_{ij}(k) \geq \alpha$ is set for safe neighbor nodes or credible attack neighbor
    nodes, and then update the information set $\Theta_i(k+1)$.
```

among its neighbors. In order to avoid infinite loop of the state, we modify the update rule for the node with minimum state which is different from the rule for the node with maximum state. When the node notices that it owns the maximum state among its neighbors, it will change its state into the second largest state. But for the node with minimum state, it will just update its state by averaging its state with the second smallest state which guarantees it will use the honest state for update and move closer to the final state.

Since for the synchronous consensus, all nodes usually have synchronized clock, it is easy to verify whether the received information is latest created. Thus, the above verification simplifies the verification of condition c_2 in (He et al. [2013]). For each safe node i, $a'_{ij}(k)$ in step 6 can be set with some simple and practical rules, for example, $a'_{ij}(k) = \frac{1}{|N_i|+1}$.

Remark 1. In (LeBlanc and Koutsoukos [2011], Zhang and Sundaram [2012], Pasqualetti et al. [2012]), when the tolerable number of attack nodes is restricted by connectivity of original network, SSCA does not need to satisfy that restriction since design of algorithm has no direct relationship with the connectivity of safe nodes. Under SSCA, attack nodes cannot attack consensus with random states selection.

4. PERFORMANCE ANALYSIS OF SSCA

In this section, we first give the matrix description of the dynamic of SSCA, and then prove the convergence property of our algorithm based on matrix theory. In the following part, regarding the states of attack nodes, we will investigate that how attack behaviors affect SSCA. Invalid and valid attack are also discussed to analyze the variation of max-min state deviation of all safe nodes under SSCA.

4.1 Matrix Description

Since under SSCA attack nodes should send credible information (can go through the verification process) to avoid being detected by the safe neighbor nodes, we assume that the information used for attack is always credible. Consider attack nodes always exist in the network.

For safe node i, it follows from steps 5 and 6 in SSCA that

$$x_i(k+1) = \sum_{j \in N_i} a'_{ij}(k) x_j(k) + a'_{ii}(k) \tilde{x}_i(k), \tag{3}$$

where $\sum_{j \in N_i} a'_{ij}(k) + a'_{ii}(k) = 1$ and $a'_{ij}(k) \geq \alpha$ hold for $j \in N_i \bigcup i$, $\tilde{x}_i(k) = \frac{x_{i_m}(k)+x_i(k)}{2}$ when $x_i(k) < x_{i_m}(k)$, $\tilde{x}_i(k) = x_{i_M}(k)$ when $x_i(k) > x_{i_M}(k)$ and $\tilde{x}_i(k) = x_i(k)$, otherwise. For attack node $i, i \in V_a$, its credible state $x_i(k+1)$ is bounded by the states of safe nodes at time k, i.e., $h'(k) \leq x_i(k+1) \leq H'(k)$, which means that there exists $a'_{ij}(k), j \in N_i \cap V_s$ to make the following equation hold,

$$x_i(k+1) = \sum_{j \in N_i} a'_{ij}(k) x_j(k).$$

Based on the above observations, there exists a row stochastic matrix $A'(k)$ such that

$$X(k+1) = A'(k) X(k). \tag{4}$$

From (3), the details of the above matrix are given as

$$\begin{aligned} \begin{bmatrix} X^s(k+1) \\ X^a(k+1) \end{bmatrix} &= \begin{bmatrix} B(k) & C(k) \\ E(k) & \mathbf{0}^{m \times m} \end{bmatrix} \begin{bmatrix} P_1(k) & P_2(k) \\ \mathbf{0}^{m \times n} & I^{m \times m} \end{bmatrix} \begin{bmatrix} X^s(k) \\ X^a(k) \end{bmatrix} \\ &= \begin{bmatrix} \tilde{B}(k) & \tilde{C}(k) \\ \tilde{E}(k) & \tilde{F}(k) \end{bmatrix} \begin{bmatrix} X^s(k) \\ X^a(k) \end{bmatrix} \end{aligned} \tag{5}$$

where $\tilde{B}(k) = B(k) \times P_1(k) \in R^{n \times n}$, $\tilde{C}(k) = B(k)P_2(k) + C(k) \in R^{n \times m}$, $\tilde{E}(k) = E(k)P_1(k) \in R^{m \times n}$, $\tilde{F}(k) = E(k)P_2(k) \in R^{m \times m}$, $P_2(k) \in R^{n \times m}$ and $I^{m \times m}$ is an identity matrix. Meanwhile, each element $b_{ij}(k) \geq \alpha$ of $B(k)$ iff $(i, j) \in E_s$ or $j = i$. $P(k) = [P_1(k) \;\; P_2(k)]$ is a matrix with its elements as follows

1. when $x_i(k) > x_{i_M}(k), i_M = j, p_{ij}(k) = 1, i \in V_s, j \in V$
2. when $x_i(k) < x_{i_m}(k), i_m = j, p_{ij}(k) = 1/2, p_{ii}(k) = 1/2, i \in V_s, j \in V$
3. otherwise, $p_{ij}(k) = 0, i \in V_s, j \in V$

The above description reveals the relationship between the states of safe nodes and attack nodes for SSCA. It is observed that some safe nodes' states are constrained by attack nodes' credible states while attack nodes' states totally depend on safe nodes' states referring to (5) among two hop information. We will analyze the performance of SSCA and attack's impact on SSCA in the following part based on these tools.

4.2 Convergence of the Algorithm

We give a theorem to guarantee convergence with an exponential rate of SSCA and then investigate when attack can accelerate consensus under SSCA.

Theorem 1. Under SSCA, convergence of discrete consensus is achieved with an exponential rate in the presence of attack, i.e., there holds that,

$$\lim_{k \to \infty} x_i(k) = c, \forall i \in V_s$$

where c is a constant.

Proof. According to (5), we have

$$\begin{bmatrix} X^s(k+1) \\ X^a(k+1) \end{bmatrix} = \begin{bmatrix} \tilde{B}(k) & \tilde{C}(k) \\ \tilde{E}(k) & \tilde{F}(k) \end{bmatrix} \begin{bmatrix} X^s(k) \\ X^a(k) \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{B}(k)\tilde{B}(k-1) + \tilde{C}(k)\tilde{E}(k-1) & \tilde{B}(k)\tilde{C}(k-1) + \tilde{C}(k)\tilde{F}(k-1) \\ \tilde{E}(k)\tilde{B}(k-1) + \tilde{F}(k)\tilde{E}(k-1) & \tilde{E}(k)\tilde{C}(k-1) + \tilde{F}(k)\tilde{E}(k-1) \end{bmatrix}$$

$$\begin{bmatrix} X^s(k-1) \\ X^a(k-1) \end{bmatrix}$$

$$= \begin{bmatrix} W^1(k) & C^1(k) \\ E^1(k) & W^2(k) \end{bmatrix} \begin{bmatrix} X(0) \\ \mathbf{0}^{m\times 1} \end{bmatrix}.$$

where $W^1(k) \in R^{n\times n}$, $C^1(k) \in R^{n\times m}$, $E^1(k) \in R^{m\times n}$ and $W^2(k) \in R^{m\times m}$. Denote

$$W(k) = A'(k)A'(k-1)...A'(0)$$
$$= \begin{bmatrix} W^1(k) & C^1(k) \\ E^1(k) & W^2(k) \end{bmatrix}$$

Since each $A'(k)$ is a row stochastic matrix, we get row stochastic matrix $W(k)$. Consider that node q, $q \in V_s$ has the largest initial state and more than two nodes in neighborhood. Since all safe nodes are always strongly connected with each other, all other safe nodes will get node q's state $x_q(0) = H'(0)$ after $T = n$ iterations for their states update. The detailed analysis is as follows.

Because $x_q(0)$ is the largest initial state, under our algorithm node q will update its state as follows $x_q(0) = x_{q_M}(0) \geq h'(0)$. For its one hop safe neighbor node i_1, if $x_{i_1}(0) < x_{i_1 m}(0)$, node i_1 will update its state by setting the weight $a'(i_1 q) \geq \alpha$ for node q. For node q with $x_q(1)$, when $x_q(1) > x_{q_M}(1)$, it will firstly update its state as follows $x_q(1) = x_{q_M}(1) \geq h(1)$. For its two hop safe neighbor node i_2, if $x_{i_2}(0) < x_{i_2 m}(0)$, node i_2 will update its state by setting the weight $a'(i_2 q) \geq \alpha^2$ for node q. For its r hop safe neighbor node i_r, if $x_{i_r}(r-1) < x_m(r-1)$, node i_n will update its state by setting the weight $a'(i_r q) \geq \alpha^r$ for node q. Therefore, after $T = n$, there must exist that $W^1(T)$ is a matrix with all elements being positive in column q and $W^1_{iq}(T) \geq (\frac{\alpha H(0)}{h(0)})^T$, and $0 < \sum_1^n w^1_{iq}(T) \leq 1$. Each safe node's state can be represented as

$$x_i(T) = \sum_{j=1}^{n} w^1_{ij}(T)x_j(0).$$

Therefore, we have $H'(T) < H'(0)$ and $h'(T) > h'(0)$ and there exists $\min[w^1_{iq}(T)] \leq \sigma_1 \leq 1 - \min[w^1_{iq}(T)]$ to make $D'(T) < \sigma_1 D'(0)$ hold.

If safe node q with the largest state among all safe nodes at iteration T, i.e., $x_q(T) = H'(T)$. If $x_q(T) > x_{q_M}(T)$, then node q will update its state as $x_q(T) = x_q(T)$ firstly. Otherwise, it will just update its state according to (2). Hence, after T iteration, just as our analysis before, there exists $\min[w^1_{iq}(2T)] < \sigma_3 < 1 - \min[w^1_{iq}(2T)]$ to make $D'(2T+1) < \sigma D'(T)$ hold. Otherwise, we consider that $x_{q_M}(T+1) = \max[X(T+1)] > H'(T+1), q_M \in V_a$ at iteration $T+1$. Since attack node q_M has to pass the verification process, its state satisfies $x_{q_M}(T+1) \leq H'(T)$. Then, attack node q_M will update its state as the second largest state of its neighbor. Hence, the state of attack node q_M will decrease. For the r hop safe neighbor node i_r of node q, it will set the weight $a'(i_r q) \geq \alpha^r$ for node q. For the same reason, there exists $\min[w^1_{i_1 q}(2T+1)] < \sigma_2 < 1 - \min[w^1_{i_1 q}(2T+1)], w^1_{i_1 q}(2T+1) \geq (\frac{\alpha H(T)}{h(T)})^n w^1_{i_1 q}(T+1)^n$ to make $D'(2T+1) < \sigma_2 D'(T)$ hold. Therefore, we have $\min[w^1_{iq}(2T+1)] < \sigma_2 < 1 - \min[w^1_{iq}(2T+1)], i \in V_s, i \in N_q$ to make $D'(2T+1) < \sigma_2 D'(T)$ hold. It infers that when $k > pT + p$, $T \geq n$, we have

$$0 \leq D'(pT) < \sigma_p \sigma_{p-1} ... \sigma_1 D'(0),$$

Hence, we have

$$\lim_{k\to\infty} D'(k) = 0,$$

which means that $\lim_{k\to\infty} x_i(k) = c, \forall i \in V_s$ and consensus is achieved with an exponential rate.

***Remark** 2.* In the proof part of Theorem 1, it is observed that attack nodes' states can be constrained in a limited interval under SSCA, which means that different states of attack nodes may bring different results. In the next subsection, related detailed analysis will be provided.

4.3 *The Effect of Attack Behavior*

Here, We first provide a condition for analysis of faster convergence under SSCA. Then, the variation of max-min state deviation is discussed for one iteration update by classifying attack into valid and invalid attack.

Theorem 2. Under SSCA, if $h'(k) \leq x_i(k) \leq H'(k), i \in V_a$, we have $X^s(k+1) = \hat{A}(k)X^s(k)$, where $\hat{A}(k)$ is a row stochastic matrix.

Proof. When $h'(k) \leq x_j(k) \leq H'(k), j \in V_a$, there exists $\omega^a_{ji}(k) \geq 0, i \in V_s$ with $\sum_{j=1}^n \omega^a_{ji}(k) = 1$, such that

$$x_j(k+1) = \sum_{i=1}^{n} \omega^a_{ji}(k)x_i(k).$$

Hence, for $\forall i \in V_s$, under SSCA, the following equation

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{n} a'_{ij}(k)x_j(k) + \sum_{j=n+1}^{n+m} a'_{ij}(k)x_j(k) \\ &= \sum_{j=1}^{n} a'_{ij}(k)x_j(k) + \sum_{j=n+1}^{n+m} a'_{ij}(k)[\sum_{q=1}^{n} \omega^a_{jq}(k)x_q(k)] \\ &= \sum_{j=1}^{n} \hat{a}_{ij}(k)x_j(k) \end{aligned}$$

holds, where $\sum_{j=1}^{n+m} a'_{ij}(k) = 1$. Note that

$$\sum_{j=1}^{n} \hat{a}_{ij}(k) = \sum_{j=1}^{n} a'_{ij}(k) + \sum_{j=n+1}^{n+m} a'_{iq}(k)(\sum_{q=1}^{n} \omega^a_{jq}(k)) = 1.$$

Therefore, there exists a row stochastic matrix $\hat{A}(k)$ such that

$$X^s(k+1) = \hat{A}(k)X^s(k).$$

***Remark** 3.* Through Theorem 3, when the states of attack nodes are between the maximum and the minimum state of all safe nodes' states, the system can be noted as linear time-varying system with zero-input. However, $\hat{A}(k)$ is complex and is hard to obtain, because it is decided by many factors, such as value vector of safe nodes, attack nodes' states selection, the weight setting mode, etc. Meanwhile, the convergence rate of a time-varying dynamic consensus algorithm is a challenging problem. Therefore, it is hard to give a condition that when the attack is beneficial for consensus.

Assume that attack nodes know the rules and obey it to attack the consensus performance of our algorithm. We give an example to illustrate attack nodes' benefit for consensus under the case described by Theorem 3 shown in Fig. 1.

Example 1: Consider a network with $n = 5$ safe nodes and $m = 1$ attack node. Set that $X(0) = [2, 3, 5, 6, 7]^T$ and attack node 6 always sends safe node 4's current state value to attack consensus. Weight setting mode is to average neighbor nodes' value with its own value. Shown in Fig. 2, consensus is reached with a faster rate. Since the appearance of attack makes the connectivity of network stronger than before under SSCA, which increases the communication frequency among safe nodes, consensus can be achieved synchronously with a faster rate. It can be observed that this 1-connected network can tolerate one attack node, which means that SSCA relaxes assumption that the tolerable number of attack is strictly decided by the connectivity of network in (Pasqualetti et al. [2010], Zhang and Sundaram [2012], LeBlanc and Koutsoukos [2011]).

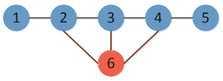


Fig. 1. An Illustrating Example

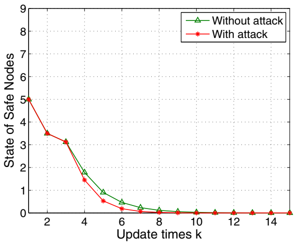


Fig. 2. Max-min deviation

The above part has pointed out that when the states of attack nodes satisfy the given condition, attack may even enhance the convergence rate of consensus. We will provide a method to classify different message manipulation attacks into invalid and valid attack such that the variation of max-min state deviation of all safe nodes for one iteration update can be fully investigated. A sufficient condition for invalid attack in a special class of consensus is obtained.

If an attack cannot increase one iteration maximum distance of all safe nodes' states, i.e, $D'(k+1) \le D(k+1)$, the attack at iteration k is defined as an invalid attack, and otherwise defined as a valid attack. Here the special class of consensus means that the weight setting mode does not change, i.e., for equations (1) and (4),

$$\frac{a'_{ij}(k)}{\sum_{j=1}^{n} a'_{ij}(k)} = a_{ij}(k), i, j \in V_s, \tag{6}$$

where $a_{ij}(k)$ is the weight that node i sets for node j without attack injection. A simple consensus rule that each node updates its state based on the average of its own state plus the states of its neighbors is widely researched. A sufficient condition for invalid attack is given as follows.

Theorem 3. Under SSCA, if $x_j(k), j \in V_a$ satisfies $h(k+1) \le x_j(k) \le H(k+1)$, we have

$$D'(k+1) \le D(k+1),$$

i.e., this attack at iteration k is an invalid attack.

Proof.

For safe node i with state $x_i(k)$ satisfying $x_{i_m(k)} \le x_i(k) \le x_{i_M}(k)$, under SSCA in the presence of attack nodes, we have

$$x_i(k+1) = \sum_{j=1}^{n} a'_{ij}(k)x_j(k) + \sum_{j=n+1}^{n+m} a'_{ij}(k)x_j(k).$$

Because of $h(k+1) \le x_j(k) \le H(k+1), j \in V_a$, we get an inequality as follows

$$\begin{aligned} x_i(k+1) &\ge \sum_{j=1}^{n} a'_{ij}(k)x_j(k) + \sum_{j=n+1}^{n+m} a'_{ij}(k)h(k+1) \\ &\ge (\sum_{j=1}^{n} a'_{ij}(k)) \sum_{j=1}^{n} \{\frac{a'_{ij}(k)}{\sum_{j=1}^{n} a'_{ij}(k)} x_j(k)\} + \\ &\quad \sum_{j=n+1}^{n+m} a'_{ij}(k)h(k+1) \end{aligned} \tag{7}$$

According to (6), we have

$$\begin{aligned} x_i(k+1) &\ge (\sum_{j=1}^{n} a'_{ij}(k)) \sum_{j=1}^{n} (a_{ij}(k)x_j(k)) + \sum_{j=n+1}^{n+m} a'_{ij}(k)h(k+1) \\ &\ge (\sum_{j=1}^{n} a'_{ij}(k))h(k+1) + \sum_{j=n+1}^{n+m} a'_{ij}(k)h(k+1) \\ &\ge h(k+1). \end{aligned} \tag{8}$$

For the same reason, we can get $x'_i(k+1) \le H(k+1)$.

For safe node i with state satisfying $x_i(k) > x_{i_M}(k)$ or $x_i(k) < x_{i_m}(k)$, under SSCA without attack, we have

$$x_i(k+1) = \sum_{j \in N_i} a_{ij}(k)x_j(k) + a_{ii}(k)x_{i_M}(k)$$

or

$$x_i(k+1) = \sum_{j \in N_i} a_{ij}(k)x_j(k) + a_{ii}(k)(x_{i_m}(k) + x_i(k))/2.$$

Since $h(k+1) \le x_i(k+1) \le H(k+1)$, we have $h(k+1) \ge x_{i_m}(k)$ and $x_{i_M}(k) \ge H(k+1)$.

If there exist attack nodes in its neighbor at update and their states satisfy $h(k+1) \le x_j(k) \le H(k+1), j \in V_a$, the following equality holds

$$x_i(k+1) = \sum_{j \in N_i \cap V_s} a'_{ij}(k)x_j(k) + a'_{ii}(k)x_{i_M}(k) + \sum_{j \in N_i \cap V_a} a'_{ij}(k)x_j(k)$$

or

$$\begin{aligned} x_i(k+1) = &\sum_{j \in N_i \cap V_s} a'_{ij}(k)x_j(k) + a'_{ii}(k)\frac{(x_{i_m}(k) + x_i(k))}{2} + \\ &\sum_{j \in N_i \cap V_a} a'_{ij}(k)x_j(k). \end{aligned}$$

According to the same reason as (7) and (8), we get $h(k+1) \le x_i(k+1) \le H(k+1)$.

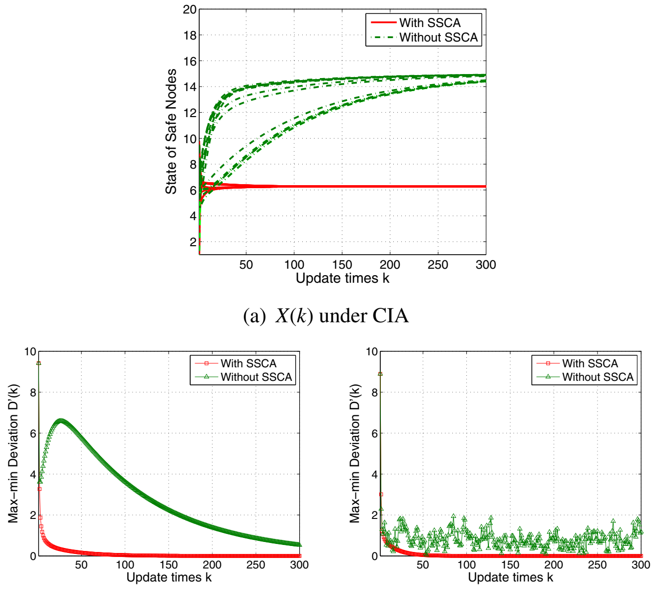
Therefore, if $x_j(k), j \in V_a$ satisfies $h(k+1) \le x_j(k) \le H(k+1)$, we have $H'(k+1) \le H(k+1)$ and $h'(k+1) \ge h(k+1)$. Hence, $D'(k+1) \le D(k+1)$ holds.

When $x_i(k) \in [x_{im}(k-1), x_{iM}(k-1)], \forall i \in V_a$, attack node i will select $x_{im}(k-1)$ or $x_{iM}(k-1)$ to attack consensus intuitively to make $D'(k+1)$ as large as possible. On the other hand, for safe nodes, the information they get is expected to come from safe nodes, which means that the information is updated by obeying SSCA. Hence, attack nodes should send information which is deviated from the real one to realize valid attack.

5. EVALUATION

In this part, we provide extensive simulation results to testify the obtained theoretical results. Consider a network with $n = 19$ safe nodes and $m = 1$ attack node, and safe nodes' states are selected from the interval $[0, 10]$ randomly. We first investigate how the network equipped with SSCA evolves under constant value attack. Specifically, we let the attack node i broadcast $x_i(k) = 15$ constantly, and draw the system state in Fig. 3(a) and Fig. 3(b). It can be seen that SSCA effectively avoids that final state of consensus is seriously deviated from the true value and consensus is achieved with an exponential convergence speed. The reason is that under SSCA, the states of attack nodes are effectively bounded by their safe neighboring nodes, which therefore guarantees that the final state of the whole network will not deviate from the true value. Instead, without SSCA, although the max-min deviation of the whole network will still decrease, the converging speed is much slower which can be seen through Fig. 3(b).

Then, We investigate how the network equipped with SSCA evolves under random value attack. Specifically, we let the attack node i broadcast $x_i(k) \in [0, 10]$ randomly, and draw the system state in Fig. 3(c). It can be seen that SSCA ensures that consensus is achieved with an exponential convergence speedWithout SSCA, in a short time, the max-min deviation of the network will remain fluctuating around as manipulated random information attack keeps injecting into the system.



(a) $X(k)$ under CIA

(b) $D'(k)$ under CIA

(c) $D'(k)$ under RIA

Fig. 3. Performance of SSCA

6. CONCLUSION

In this paper, we mainly consider secure consensus in synchronous networks under message manipulation attacks. Based on two-hop neighbor nodes' information, we first propose SSCA by revising the existing SATS protocol (He et al. [2013]) which was proposed for consensus in asynchronous sensor networks. By utilizing matrix description of the nodes' states dynamics, we prove that SSCA can guarantee the exponential convergence even under message manipulation attacks. We further analyze how the attack behaviors affect the convergence of consensus. Such result is also exploited for classifying the attack behavior and analyzing the convergence rate. Illustrating examples and extensive simulation results verify the effectiveness of proposed mechanism and theoretical analysis.

REFERENCES

Cao, X., Chen, J., Zhang, Y., and Sun, Y. (2008). Development of an integrated wireless sensor network microenvironmental monitoring system. *ISA Trans*, 47(3), 247–255.

He, J., Chen, J., Cheng, P., and Cao, X. (2014a). Secure time synchronization in wirelesssensor networks: A maximum consensus-based approach. *IEEE TPDS*, 25(4), 1055–1065.

He, J., Cheng, P., Shi, L., and Chen, J. (2013). Sats: Secure average-consensus-based time synchronization in wireless sensor networks. *IEEE TSP*, 61(24), 6387–6400.

He, J., Cheng, P., Shi, L., Chen, J., and Sun, Y. (2014b). Time synchronization in wsns: A maximum-value-based consensus approach. *IEEE TAC*, 59(3), 660–675.

Khanafer, A., Touri, B., and Basar, T. (2012). Consensus in the presence of an adversary. In *IFAC Workshop on NecSys*, 276–281.

LeBlanc, H. and Koutsoukos, X. (2011). Consensus in networked multi-agent systems with adversaries. In *Proc. ACM HSCC*, 281–290.

LeBlanc, H., Zhang, H., Koutsoukos, X., and Sundaram, S. (2013). Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4), 766–781.

Lu, R., Lin, X., Zhu, H., Liang, X., and Becan, X.S. (2012). Becan: A bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. *IEEE TPDS*, 23(1), 32–43.

N. H. Vaidya, a.L.T. and Liang, G. (2012). Iterative approximate byzantine consensus in arbitrary directed graphs. In *Proc. ACM symposium on PODC*, 365–374.

Olfati-Saber, R., Fax, J.A., and Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.

Pasqualetti, F., Bicchi, A., and Bullo, F. (2012). Consensus computation in unreliable networks: A system theoretic approach. *IEEE TAC*, 57(1), 90–104.

Pasqualetti, F., Carli, R., Bicchi, A., and Bullo, F. (2010). Distributed estimation and detection under local information. In *IFAC Workshop on NecSys*, 263–268.

Ren, W., W.Beard, R., and M.Atkins, E. (2005). A survey of consensus problems in multi-agent coordination. In *Proc. IEEE ACC*, 1859–1864.

Xiao, L., Boyd, S., and Lallo, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Proc. IEEE IPSN*, 63–70.

Zhang, H. and Sundaram, S. (2012). A simple median-based resilient consensus algorithm. In *Proc. Annual Allerton Conference on Communication, Control, and Computing*, 1734–1741.

Zhou, M., He, J., Cheng, C., and Chen, J. (2013). Discrete average consensus with bounded niose. In *Proc. IEEE CDC*, 5270–5275.