Attacker Location Evaluation-based Fake Source Scheduling for Source Location Privacy in Cyber-Physical Systems

Zhen Hong*, Rui Wang, Shouling Ji, and Raheem Beyah, *Senior Member*, *IEEE*

***Abstract*—Cyber-physical systems (CPS) have been deployed in many areas and have reached unprecedented levels of performance and efficiency. However, the security and privacy problems in CPS have not been properly addressed, e.g., the monitored source location can be inferred by an attacker, which can substantially undermine the reliability of CPS. Unfortunately, the existing techniques to protect against the leakage of the source location do not achieve an acceptable balance among source location privacy, transmission delay, and energy consumption to guarantee high reliability. To address this issue, we propose an attacker location evaluation-based fake source scheduling (FSSE) for source location privacy in CPS to enhance the privacy level and maintain the system performance. The proposed FSSE contains two main phases. The first, backbone construction, is dependent on the probability of capture derived from the communication information of self and neighboring nodes. This phase aims to build a backbone to form a baseline with respect to the source location privacy and transmission delay. The second phase is fake message scheduling, which is established to provide a tradeoff among privacy, transmission delay, and communication overhead in terms of the hypothesized location of the attacker by using stochastic processes. Through analysis and simulation, we demonstrate that the proposed method has a more stable privacy level and more efficient transmission delay and energy consumption than the three compared algorithms, i.e., phantom routing (PR), tree-based diversionary routing (TDR) and dynamic fake source selection (DFSS).**

***Index Terms*—Cyber-physical systems, source location privacy, fake source scheduling, attacker location evaluation, backbone construction, first-passage time.**

Manuscript received March 1, 2018; revised September 15, 2018; accepted September 21, 2018. Date of publication October X, 2018; date of current version October X. This work was supported in part by the National Natural Science Foundation of China under Grant No. 61304256 and 61772466, Zhejiang Province Public Technology Project under Grant No. 2017C33153, Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under Grant No. R19F020013, Provincial Key Research and Development Program of Zhejiang under Grant No. 2017C01055, Young Researchers Foundation of Zhejiang Provincial Top Key Academic Discipline of Mechanical Engineering and Zhejiang Sci-Tech University Key Laboratory under No. ZSTUME01B15, New Century 151 Talent Project of Zhejiang Province, 521 Talent Project of Zhejiang Sci-Tech University, and Young and Middle-aged Talents Foundation of Zhejiang Provincial Top Key Academic Discipline of Mechanical Engineering. (*Corresponding author: Zhen Hong)

Zhen Hong and Rui Wang were with the Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou, Zhejiang 310018 China. Zhen Hong is now with the Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou, Zhejiang 310023 China. (Email: zhong@zstu.edu.cn; wangrui_key@163.com).

Shouling Ji is with the Institute of Cyber Security Research and College of Computer Science & Technology, Zhejiang University, Hangzhou, Zhejiang 310027 China (Email:sji@zju.edu.cn).

Raheem Beyah is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (Email:rbeyah@ece.gatech.edu).

I. Introduction

CYBER-PHYSICAL systems (CPS) are sophisticated control-computing hybrids that provide unprecedented performance and efficiency. Usually, CPS refers to a new generation of systems with integrated computational and physical capabilities that can achieve interconnection between the physical world and cyberspace [1], [2]. CPS involve large-scale application domains, such as smart grids, intelligent healthcare, industrial control systems and aerospace systems [3]–[6]. Specifically, industrial control systems (ICS) which are broadly applied to smart manufacturing, are the most common CPS. Fig.1 shows a simplified industrial/process control system, which is also a kind of CPS.

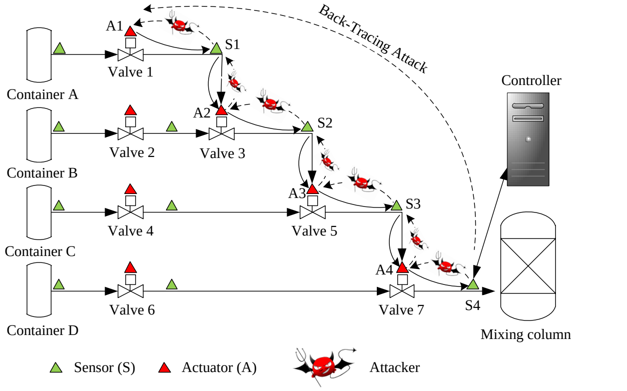


Fig. 1. A simplified case of a source location privacy attack in a CPS network.

Currently, the security and privacy of CPS is an essential issue [7]. CPS are usually deployed in closed environments that are disconnected from the public network. However, CPS can still be infected by malicious software if the internal staff of the company inserts an infected USB disk. A real case is Stuxnet, which occurs in the energy network (e.g., power grid) in the industrial world [8]. Thus, an attacker has access to the current CPS network if any node, physical device or controller is infected by malicious software. Specifically, the location of the event source can be inferred by the attacker regardless of the encryption strength.

Because many old machines are in use in factories, sensors and actuators are deployed or directly embedded in physical devices to collect data or control components. These sensors and actuators compose the CPS network (wireless or wire), in addition to the original CAN-bus between physical devices. In Fig.1, A1 (Actuator) is very important in the current CPS network since it controls Valve 1 to adjust the liquid flow from Container 1. Assume that A1 is the source in the network at one moment. We can build a path "A1→S1→A2→S2→A3→S3→A4→S4" for data forwarding. Because the attacker can always monitor the network [9], he can easily eavesdrop on the message on the path between the source and the sink (controller in Fig.1). Afterward, the attacker searches for the source from S4 to A1 using a back-tracing mechanism. Once the location of A1 in the CPS network is discovered, it may be maliciously attacked, which can result in a major disaster. Note that the main target of the attacker is to identify the location of the key node and then damage the physical devices. Due to the limited resources, real-time response, high fault tolerance and high security, CPS requires significant non-functional real-time and reliability attributes [10]. Consequently, the preservation of source location privacy in CPS is quite challenging. A privacy-aware routing protocol is urgently needed to achieve a high level of location privacy and service quality.

Recently, a large number of privacy-preserving routing techniques, which mainly adopt mechanisms of flooding, random walk, network coding, fake source and dummy data, have been proposed [11], [12]. Ozturk et al. [13] propose a flooding and single-path routing protocol, namely, phantom routing (PR), which aims to entice the attacker away from the source. In PR, the delivery of every message experiences two phases, i.e., a random walk and a subsequent flooding/single-path routing. However, PR cannot provide stable privacy due to the uncertainty of the random walk. Generally, the random walk increases the transmission delay, and the flooding routing aggravates the transmission overhead. Mehta et al. [14] present a scheme of a fake source deployed for location protection, in which one or more virtual sources are simulated to confuse the attacker. However, the only information that is missing is the exact number of fake sources under random selection, which may cause asymmetrical distribution, energy holes, and an unstable privacy level.

In this paper, we focus on privacy-aware routing using a fake source mechanism in the CPS network. To improve the performance and privacy of routing in most existing studies, we propose an attacker location evaluation-based fake source scheduling (FSSE) algorithm to address the source location privacy in CPS. The FSSE consists of two parts: backbone construction and fake message scheduling. First, backbone construction builds a path from the source to the sink to deliver the message while considering the number of adjacent nodes to satisfy the given captured threshold. Subsequently, fake message scheduling selects nodes as the fake source in terms of their states and probable attacker position, in which the states include the residual energy and the historical information of neighbors. Specifically, the probable attacker position is evaluated by using stochastic process theory. Combined with the probability obtained by the abovementioned information, the node in the backbone can decide whether to be the fake source and its duration via a stochastic strategy. Consequently, the constructed backbone provides stable privacy compared with the random walk mechanism, while fake message scheduling establishes a distributed source location protection mode and achieves a trade-off among privacy, latency, and overhead.

The contributions of this paper are summarized as follows.

- Inspired by the distributed topology control, we propose an attacker location evaluation-based fake source scheduling (FSSE) method using stochastic processes theory. The first phase of FSSE (i.e., backbone construction) ensures the baseline level of privacy and efficient transmission, while the second phase (i.e., fake message scheduling) is a distributed mechanism for selecting the fake source to broadcast a one-hop fake message confuse the attacker. A trade-off among location privacy, transmission delay, and communication overhead is achieved to optimize the source location privacy and the network performance.
- We use stochastic process theory to evaluate the attacker location and model the motion of the attacker, whose motion model is proved to satisfy the Markov property. Then, we formulate the state transition relationship of the attackers and propose a distributed probability-based fake message scheduling algorithm while considering the state of the node.

The rest of this paper is organized as follows. In section II, we discuss the related work regarding the existing approaches to preserving source location privacy. The notations and assumptions, as well as the system model and the attacker model, are given in section III. We formalize the source location privacy problem in CPS in section IV and present the proposed FSSE for preserving source location privacy in detail in section V. Subsequently, the performance of the proposed FSSE is compared with that of other typical algorithms via a simulation study in section VI. Finally, we conclude this paper in Section VII.

II. Related work

PR is a typical scheme for preserving source location privacy in a network. The random walk, which can easily cause the issue of unstable privacy, is a common method adopted in PR. To address this problem, Kamat et al. [15] propose a directed random walk mechanism that makes the attacker be far from the real source with a greater probability. Furthermore, Li et al. [16] and Kumar et al. [17] propose multiple phantom nodes selected based on a directed random walk, in which the characteristics of distance, angle, and phase between nodes are considered to determine the candidate relay nodes. Indeed, this method provides more stable privacy than the previously pure random walk mechanism, but it increases the hardware requirements and additional overhead.

Tree-based diversionary routing (TDR) is proposed in [18] for improving location privacy and ensuring the network lifetime. The main idea is to construct a backbone with branches that contains both the real source and the phantom node, and at the ends of the other branches are fake sources. Due to the homogeneous property of the established tree branches, the attacker cannot infer the real source location from the transmitted messages and the features of the network architecture. Nevertheless, the nodes must be location-aware, and their constructed tree is complicated, which means the fake source may not be extended to the network edge. Meanwhile, the dummy message mechanism is established by idle nodes to entice the attacker, but it unintentionally increases the communication overhead and transmission delay. Moreover, the energy consumption and transmission delay are worse than those of PR because the constructed backbone is not the shortest path.

To solve the issue of higher energy consumption and transmission delay in TDR, Jhumka et al. [19] formalize the fake source selection as an NP-Complete problem while proposing a heuristic algorithm called distributed fake source selection algorithm. According to the local information (i.e., historical state and state exchange message), each node determines whether to be the fake source and then floods fake messages periodically for the duration. From [19], we can see that broadcasting fake messages with high frequency would enhance the privacy but aggravate message collision. Thus, the privacy and the network performance depend on the duration of the fake source and the flooding frequency. Furthermore, Thomason et al. [20] point out that increasing the broadcast rate of fake messages would increase the probability of message conflict, which would also indirectly reduce the source location privacy. Therefore, Jhumka et al. [21] and Bradbury et al. [22] propose an improved algorithm, i.e., an adaptive dynamic fake source selection (DFSS), which provides a reasonable flooding duration and frequency for fake messages. DFSS helps to reduce the upper limit of transmission latency and enhance the fake source privacy and network performance. However, DFSS needs to flood the fake message several times, which means the relay nodes will forward multiple fake messages several times in one sample period, resulting in ten-fold greater energy consumption.

The evolution of the attacker model and the game between the attacker and the privacy protection promote the development of source location privacy studies. In [15], two types of human behavior are simulated as the attacker model, namely, patient adversary and cautious adversary. The patient adversary performs better than the cautious adversary in PR. Nevertheless, due to the immediate backtracks when listening to a new message, there are several disadvantages for the patient adversary. (1) It is sensitive when first listening to the message, and the network knowledge is limited by the local information. (2) It lacks a coping strategy for source privacy protection as well as feedback and improvement for the attack. Consequently, privacy-aware routings based on single or multi-paths can defend against the patient adversary efficiently [23].

Compared with the limited-knowledge local attacker, more threats exist in source location privacy preservation by a global eavesdropper. To address the threats, Mehta et al. [14] and Hu et al. [24] develop a solution for source location protection from a network-wide perspective to fully deploy the security mechanism across the network. These methods preserve source location privacy against the global eavesdropper at the cost of considerably reduced energy conservation. Therefore, a trade-off between privacy and communication overhead in resource-limited systems must be achieved.

Additionally, the global eavesdropper usually pays a higher cost for deploying a large-scale network to intercept. Thus, it is reasonable for a resource-limited adversary to establish an optimal attack scheduling to maximize the attack effect [25]. However, to the best of our knowledge, minimal research on the optimal attack has been performed. Generally, a traditional eavesdropper determines the direction of backtracking to be close to the source location by analyzing the network traffic. To address such an attack, Fan et al. [26], [27] propose a network-coding-based protocol. The packet is divided into several parts to be transferred to the sink along multiple paths and is then reconstructed. Traffic-analysis-based transmission into groups could efficiently defend against the attacker, but it may result in congestion and complicated data reconstruction. Moreover, as far as the attacker is concerned, a directed mobility strategy (i.e., backtracking) is adopted to get closer to the data source by traffic analysis [18], which can further help to obtain source location privacy.

In summary, PR and its improved methods sacrifice a small amount of network performance to provide better source location privacy. However, the provided privacy is relatively unstable because of the randomness of the transmission path. These methods cannot defend against multiple attackers or a global eavesdropper effectively. The fake source mechanism established by a certain selection method can confuse the backtracking path of the global attacker by flooding dummy data. This process clearly ensures the source location privacy, but flooding fake messages at high frequency considerably degrades the network performance. Therefore, modeling a real attacker and improving the diffusion pattern of fake messages may provide a better solution to enhance the performance of the fake source scheduling mechanism.

III. System model

A. Notations and Assumptions

In Table I, we list some frequently used notations, and their specific meanings are further explained in the discussion below. In addition, several assumptions are given as follows.

1) The source can be either a sensor node or an actuator.
2) Each device is equipped with a global clock for time synchronization. The time overhead, other than communication between nodes, can be ignored.
3) Each node communicates with neighboring nodes in radio frequency. The communication radius is the same for each node. There is no additional overhead or latency caused by abnormal conditions, such as packet loss, message resending and collision.
4) The network is initialized to be connected and bidirectional, and a message can be flooded to any other node.
5) The impact of the network topology construction on the attacker can be ignored. The time spent and energy consumed for topology construction are negligible.
6) All the attacker's behaviors, such as monitoring, localization, and backtracking, can be implemented within a sampling period.

B. System framework

CPS is the core of networked control systems such as industrial control and supervisory control and data acquisition

TABLE I
FREQUENTLY USED NOTATIONS

Notaion	Description
V	a set of nodes
v_i	node i, $v_i \in V$
B	a set of backbones, $B \subseteq V$
F	a set of fake sources, $F \subseteq V$
I_i	the neighbors of node i
$e_{i,j}$	the link between v_i and v_j, $e_{i,j} \in \{0,1\}$
$T^{(i)}$	period i
T^{max}	the given maximum period
T_{safe}	the safety period
T_R	the transmission delay
$T_R^{(i)}$	the transmission delay in period i
E_i	the energy consumption
E_i^j	the energy consumption in period j

TABLE II
A CPS MODEL WITH PERIODICAL SAMPLING

$T^{(i)}$	t_0	t_1	...	$t_{T^{max}}$
Source	a_0	a_1	...	$a_{T^{max}}$
Routing	r_0	r_1	...	$r_{T^{max}}$
Attacker	h_0	h_1	...	$h_{T^{max}}$
Fake source	f_0	f_1	...	$f_{T^{max}}$

[28]. In this paper, we focus on the source location privacy problem in a context of the wireless networked control system with delay. Fig.2 shows the system framework of CPS for source location privacy, which can be abstracted from Fig.1.

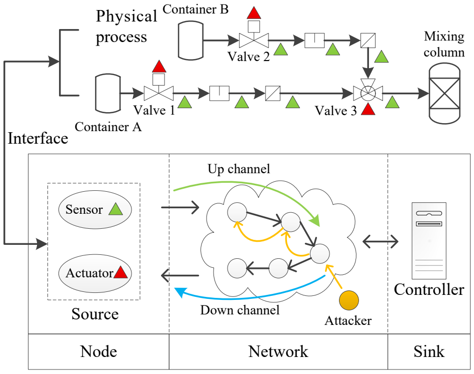


Fig. 2. The system framework of CPS for source location privacy.

As shown in Fig.2, CPS is an event-driven, discrete-sampling system that includes nodes, the network and the sink. Nodes are divided into sensor nodes and actuators: the sensor nodes are used for gathering and transmitting data and the actuators are used for receiving and performing instructions. The network consists of wireless channels that exchange messages according to routing rules. The sink, which aggregates data and sends control instructions, is the control center of the whole system.

In the context of discrete sampling systems, we establish a data-driven periodic sampling model. As shown in Table II, the clock is discretized into T^{max} uniform time periods. During each time period, all processes, including the behaviors of the normal nodes and the attackers and message transmission, are performed only once. In general, the time required to transmit each message to the sink is less than a single sampling period. Therefore, each message transmission is independent, with no mutual interference in terms of latency and overhead.

C. CPS network model

1) Network model: The CPS network is composed of nodes that can communicate with each other. Usually, each node has its own identity and some computation capability. Nodes that can communicate directly are called adjacent nodes.

2) Transmission delay: Let $T_{i,j}(l,d)$ denote the transmission delay between nodes i and j when node i sends an l bit packet to node j at distance d. The delay on link $e_{i,j}$ is given by the following formula that considers antenna gain, antenna height, system loss factor, etc. [29].

$$T_{i,j}\left(l,d\right) = \frac{l \cdot t_{i,j}}{1 - \exp\left(-0.5\gamma_{i,j}\right)} \tag{1}$$

where $t_{i,j}$ is the receiving and processing time required for each bit of data without interference, $\gamma_{i,j}$ is the signal-to-interference-noise ratio which is related to noise from the environment and can be estimated by

$$\begin{aligned} \gamma_{i,j} &= \frac{p_{i,j}^r}{p_e + \sum_{k \in I_j, k \neq i} p_{k,j}^r} \\ \text{s.t.} \quad p_{i,j}^r &= \begin{cases} \frac{g_{i,j} \cdot p_{i,j}^t}{d^2}, & d < d_0 \\ \frac{g_{i,j} \cdot p_{i,j}^t}{d^4}, & d \geq d_0 \end{cases} \end{aligned} \tag{2}$$

where $p_{i,j}^r$ represents the signal power of the message sent from node i to node j, which is related to wireless device $g_{i,j}$ and transmitting power $p_{i,j}^t$. p_e is the noise power around node j, and I_j represents the set of neighbor nodes of node j.

To simplify the transmission delay, the link quality and message resending are not considered. The transmission delay T_R from the source to the sink can be formulated as a superposition of the single-hop communication delay on transmission path (R) as follows:

$$T_R = \sum_{e_{i,j} \in R} T_{i,j}\left(l,d\right) \tag{3}$$

3) Communication overhead: We adopt the energy consumption model in [30]. The energy dissipation of a node for sending an l bit message a distance d and receiving the message are, respectively, defined as

$$E_t\left(l,d\right) = \begin{cases} l \cdot E_{elec} + l \cdot \varepsilon_{fs} d^2, & d < d_0 \\ l \cdot E_{elec} + l \cdot \varepsilon_{mp} d^4, & d \geq d_0 \end{cases} \tag{4}$$

$$E_r\left(l,d\right) = l \cdot E_{elec} \tag{5}$$

E_{elec} is the energy dissipated per bit to run the transmitter or the receiver, which depends on factors such as the digital coding, modulation, filtering, and spreading of the signal. ε_{fs} and ε_{mp} are the amplifier parameters, which depend on the distance to the receiver and the acceptable bit-error rate.

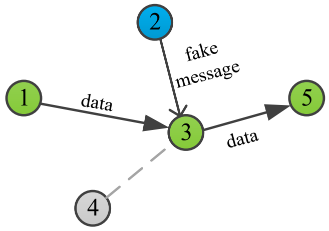


Fig. 3. The difference between transmitting sampling data and fake messages.

D. Routing rule

The routing is performed according to a rule of communication between nodes, which builds a path from the source to the sink based on the current topology. With the synchronous clock and the ideal transmission schedule, there is no idle time spent while the source sends data to the sink. In this paper, we assume that fake messages are broadcast only to neighbors and are not transmitted to the sink. For example, in Fig.3, there are five nodes in the network, where the blue node 2 and the gray node 4 are the fake source and the dormant node in this sampling interval, respectively. Note that Fig.3 is only a part of a CPS network and shows only the current data flow at a single moment. Relay node 3 receives real data from node 1 and a fake message from node 2 and then broadcasts the real data to node 5 to complete the relay task. Thus, the fake message starts at node 2 and ends at its neighbor node 3. By contrast, the real data from node 1 can continue being sent to the sink through node 5. The system feedback is transmitted by the same path, from sink to source, when the controller completes the real data.

E. Attacker model

An attacker in the network can use the surrounding wireless signal to infer and trace back to the node location of the broadcast message. Inspired by knowledge of existing works [11], we propose a simple attacker model. Assume that the attacker has no control and interferes with every node; simultaneously, the attacker also cannot verify the authenticity of eavesdropped data. The only thing the attacker can do is monitor the links between neighboring nodes located at the current node's position to infer the location of the signal source. The backtracking iterates until the attacker finds the location of the source.

Algorithm 1 The movement of the attacker

```
Require: $T^{max}$
Return: Source location
for $i = 1$ to $T^{max}$ do
    Listen(node in neighbors)
    messages = ReceiveMessages()
    nextLocation = InferImmediateSender(messages)
    MoveTo(nextLocation)
    if IsSource(nextLocation) then
        return nextLocation
    end if
end for
```

Algorithm 2 InferImmediateSender

```
Require: messages
Return: location
initialize a list LJ of location and probability
for message in messages do
    location = Positioning(message)
    LJ.append(location, random())
end for
return location with largest probability in LJ
```

As described in Algorithm 1, we use a random walk [31] as the backtracking mode of the attacker. In each sampling period, the attacker listens to the surroundings and determines the next location. Specifically, InferImmediateSender() is an equiprobability-based function used for inferring the location of the message source. Its pseudocode is given in Algorithm 2. Afterwards, the attacker evaluates whether the relay node is real and then moves to the next node.

IV. Source location privacy in CPS

A. Optimization objective

In this paper, we focus on the privacy-aware algorithm while considering the high real-time requirements and the minimum energy consumption of the system. The objective of our method can be regarded as achieving a trade-off among source location privacy, transmission delay, and average network energy consumption.

1) Source location privacy: The safety period, T_{safe}, is the number of periods before the source node is identified by the attacker. Clearly, the worst case for the attacker is to capture the source after traversing the entire network. Therefore, the safety period is generally limited. We assume that the data source is generated so frequently that the attacker can only backtrack once in a sampling period. Thus, the safety period can be maximized as

$$\max T_{safe} \tag{6}$$

where $T_{safe} \leq T^{max}$, T^{max} is the potential maximum safety period which is a constant. In the ideal scenario, the safety period can be set to a large value without considering node failure. However, a high maximum safety period may result in a large amount of computational overhead. Therefore, combined with the node's lifetime, the boundary of the safety period (i.e., T^{max}) is determined by enumeration with the minimum simulation time to ensure the value is as large as possible.

2) Transmission delay: Different routing protocols have different delays because of the different MAC protocols and hop counts, as well as the transmission loss and abnormality caused by the outer environments. We discuss the delay caused by message transmission during the safety period, i.e., the time from the start of the route to the capture of the source. To minimize the transmission delay, we have

$$\min T_R = \min \max_j T_R^{(j)}, j = 1, 2, \cdots, T_{safe} \tag{7}$$

where $T_R^{(j)}$ is the delay in period j on path R, and its maximum value is used for the evaluation metric of the transmission delay of the routing protocol. In this paper, to simplify the transmission, we do not consider the additional delay caused by the link quality and the retransmission mechanism.

3) Energy consumption: The energy consumption of the node is constrained by the battery, which means battery exhaustion impacts communication in the network and degrades its coverage. Thus, the privacy-aware algorithm should reduce the additional overhead to prolong the network lifetime. We denote E_i as the maximum average overhead per period when the first battery runs out of power. The objective of minimizing energy consumption can be expressed as

$$\min E_i = \min \frac{1}{T_{safe}} \max_{1\le i\le |V|} \sum_{j=1}^{T_{safe}} E_i^{(j)} \tag{8}$$

where $E_i^{(j)}$ refers to the overhead of node i in period j that results from data processes such as broadcasting, receiving, and aggregation. Specifically, the analysis and processing of fake messages are regarded parts of data fusion.

Consequently, the final main goal of the privacy-aware algorithm can be summarized as

$$\text{Multi-Objective} = \begin{cases} \max T_{safe} \\ \min T_R = \min \max_j T_R^{(j)} \\ \min E_i \end{cases} \tag{9}$$
$$\text{s.t.} \quad j = 1, 2, \cdots, T_{safe}$$

B. Fake source scheduling

The fake source is chosen based on the message transmission path to enhance the privacy of the traditional routing protocol; however, it generates additional delay and energy consumption. To address these issues, we include a fake source scheduling mechanism in the privacy preservation process. Fig.4 shows the fake source scheduling model for source location privacy in CPS. There are 5 nodes in the network, and the attacker and data source are located on nodes 0 and 3, respectively. The rest of the nodes (i.e., nodes 1, 2, and 4) are fake source candidates that can be chosen to broadcast fake messages to confuse the attacker. For example, when node 3 sends a message, in Fig.4, in period t_1, nodes 2 and 4 become fake sources and broadcast fake messages. Meanwhile, the attacker can receive all the signals from nodes 2, 3 and 4. Since the attacker cannot identify whether the data are real without the access records of the message source location, it uses a random strategy to move to fake source node 2. In this case, the model successfully achieves the goal of enticing the attacker by broadcasting fake messages. Therefore, combined with the case study, the problem of the fake source scheduling can be defined as how to select a node when broadcasting fake messages at any moment. As for the strategy of broadcasting fake messages, we design an algorithm by modeling the behavior of the attacker.

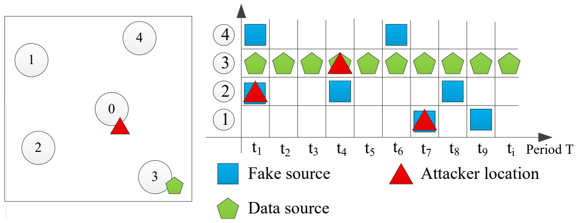


Fig. 4. An overview of fake source scheduling.

The strategy set of the fake source scheduling can be divided into the backbone (B) and fake sources (F). The backbone consists of the real source, the relay node and the sink, and the fake sources cannot cover the backbone, i.e., $F \subseteq V - B$. Thus, the objective of the privacy-aware algorithm can be formulated as

$$\text{Multi-Objective} = \text{Multi-Objective}(B, F) \tag{10}$$

In Eq. (10), we define a multi-objective integer programming model with respect to B and F for the SLP problem. The solution space is huge when the number of nodes ($|V|$) and the number of sampling periods (T^{max}) are large. In such a scenario, a long time is required to obtain the optimal solution since the calculation complexity is high.

V. The Proposed Privacy-Preserving Technique

In this section, a novel fake source scheduling algorithm (FSSE) is discussed for solving the proposed multi-objective integer programming model to achieve source location privacy and ensure low transmission delay and energy consumption. The FSSE algorithm consists of two main phases, namely, backbone construction and fake message scheduling.

A. Backbone construction

Backbone construction aims to build a transmission path between the source and the sink, which directly influences the source location privacy and the network performance. Usually, the attacker can quickly infer the source location from a short path, which results in low privacy. By contrast, a long path provides better privacy but also result in an additional latency and overhead. Therefore, how to design the length of the transmission path while ensuring sufficient idle nodes on the path to act as fake sources is the main problem to be solved in backbone construction. This process requires a trade-offs among privacy, latency, and overhead.

Many techniques can be used to build various backbones within a connected network. As shown in Fig.5, three types of backbones are derived from the connected network: the first backbone is composed of nodes 6 and 7, the second backbone includes nodes 7, 6, 4 and 5, and the third backbone includes nodes 6 and 7. The first backbone has the least hops, and the second backbone has the most hops. Assume that node 1 is the source, node 7 is the sink, and the attacker is initially located at the sink. From the perspective of transmission delay, the longer the path is, the higher the delay of the backbone. In Fig.5, the second backbone has the largest latency, but it also provides the best privacy because the attacker has a 1/24 probability of capturing the source by randomly selecting a relay node with equal probability in the backbone. By contrast, the first backbone provides the worst privacy because the attacker has a 1/6 probability of capturing the source. The third backbone is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2018.2876839, IEEE Transactions on Information Forensics and Security

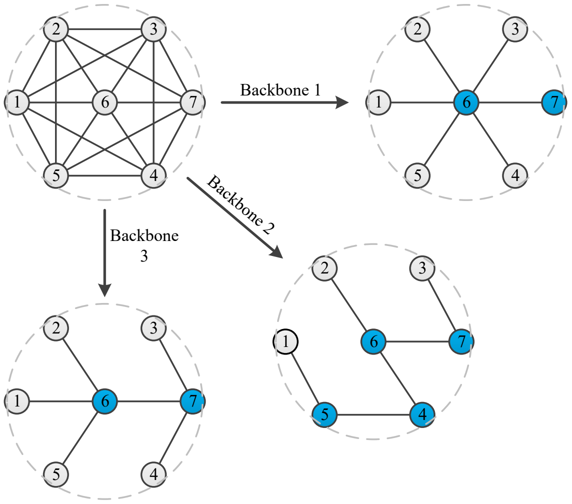


Fig. 5. Various backbones derived from a connected network.

a better choice due to the 1/12 capture probability. Therefore, the expected probability of being captured, namely, $E(B)$, can be estimated by

$$\begin{aligned} E(B) &= \prod_{b_i \in B-A} E(b_i) \\ &= \prod_{b_i \in B-A} \frac{1}{|I_{b_i}|} \end{aligned} \tag{11}$$

where b_i is a node among the backbone other than the source (A), and I_{b_i} is the neighbors of node b_i. According to (11), we propose a path search method for minimizing the source capture probability of the backbone, as depicted in Algorithm 3.

Algorithm 3 Backbone construction

```
Require: A connected network $G$, a source $A$, a sink $S$, a likelihood
    threshold $C_E$, a max step $L^{max}$.
Return: A backbone connected source and sink.
Queue.push(($S$, level $= 0$))
while Queue is not empty do
    u, level = Queue.pop()
    u broadcasts (id, level) within its radius
    for $v_i$ in nodes which got the (id, level) do
        $v_i$ updates its neighbors table
        if $v_i$ is not visited then
            Queue.push(($v_i$, level $+1$))
        end if
    end for
end while
repeat
    target, former = searchDeepFirst($u$, former, $S.level$, 1.0)
    $L^{max}$ += 1
until sink.likelihood $\leq C_E$
return former
```

In Algorithm 3, as inspired by the breadth-first search method, the message broadcast starts at the sink. Subsequently, the message, including the id and the level, is flooded to the other nodes, where the level is the least hops to the sink. When the node receives the message, it calculates the hops to the sink using the level value in the message. After constructing the network hierarchy (i.e., tree topology), as specified in Algorithm 4, we use the depth-first search to find a path from the source to the sink, which satisfies the captured probability threshold, as a channel for uploading data. During the depth-first search, the source as the start node broadcasts the construction message, including the maximum steps, traversed path and probability. When the neighboring node receives the message, it estimates the capture probability by Eq. (11) and then broadcasts the updated message. Once the sink receives the construction message, it determines the backbone path based on whether the capture is equal to or less than the given threshold. Specifically, if the minimum capture probability is larger than the threshold C_E, then the maximum length of the backbone increases. For a backbone constrained by the threshold C_E, we analyze the sensitivity of threshold C_E in the subsequent simulation.

Algorithm 4 searchDeepFirst

```
function SEARCHDEEPFIRST(u, former, $L^{max}$, target)
    u broadcast(id, $L^{max}$, u.likelihood)
    for $v_i$ in nodes which get the (id, $L^{max}$, u.likelihood) do
        if $v_i$ not visited or former.length + $v_i$.level $\leq L^{max}$ then
            $v_i$.likelihood = u.likelihood/$|I_i|$
            former.append($v_i$)
            if $v_i$ is sink then
                target = min($v_i$.likelihood, target)
                return target, former
            else
                target, newformer = searchDeepFirst($v_i$, former,
                                                    $L^{max}$, target)
                if target $\leq C_E$ then
                    return target, newformer
                end if
            end if
            former.pop()
        end if
    end for
    return target, former
end function
```

B. Fake message scheduling

The backbone construction not only satisfies the given expected probability of being captured but also achieves the goal of approximately optimal transmission delay. In general, there are always enough neighboring nodes around the backbone. According to the formula of the transmission delay, all spare nodes that are adjacent to the backbone can act as fake sources and broadcast fake messages, which enhances the privacy but aggravates the delay and does not satisfy the real-time requirement. On the other hand, if there are no additional fake sources other than the neighboring nodes of the backbone, then the attacker will backtrack to the backbone after one hop. To address the issue, we propose an optimal fake source scheduling scheme by modeling the movement of the attacker as a Markov chain to simulate the backtracking of the attacker. The movement model is used for estimating the attacker's transition between neighboring nodes and for inferring the attacker's location.

Lemma 1: The movement of an attacker walking randomly with equal link probability in a CPS network is subject to the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2018.2876839, IEEE Transactions on Information Forensics and Security

Markov property: it is a Markov chain.

Proof: Given a network, the node locations that an attacker can move to are expressed as $v_j \in X$, where X denotes the state space of the attacker and $X = \{v_1, \ldots, v_{|V|}\}$. Due to the independence between each of the attacker's backtracking steps, the transition relationship between the state space satisfies

$$\begin{aligned} P\left\{X_{i+1} = v_j | X_1 = v_{i_1}, \cdots, X_{i-1} = v_{i_{i-1}}, X_i = v_i\right\} \\ = P\left\{X_{i+1} = v_j | X_i = v_i\right\} \end{aligned}$$

Therefore, the movement of the attacker is a Markov chain subject to the Markov property. Furthermore, the state space can be expressed as

$$\begin{aligned} P &= [p_{i,j}]_{|V| \times |V|} \\ \text{s.t.} \quad p_{i,j} &= P\{X_{n+1} = v_j | X_n = v_i\} \\ &= \begin{cases} \frac{1}{|I_i|+1} & v_j \in I_i \cup \{v_i\} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (12)$$

□

Assume that the CPS network is a connected graph that has both up and down channels in all wireless links. Given an initial node i, any other nodes on the connected graph has a connected path to node i so that it can start from node i and reach node j within a certain time t. Node i is reachable to node j. Similarly, node j is reachable to node i. Therefore, node i and j are connected, which means any two nodes in the connected graph are connected.

***Lemma** 2:* The movement of backtracking mode, which aims to capture the source node, has a state space containing an absorbing state.

Proof: According to Lemma 1, the attacker's backtracking movement in the network is a Markov chain. The attacker's goal is to reach the location of the source and capture it. When the attacker reaches the source node for the first time, it will not move to another node. Therefore, the source location is a closed set that has a single state, i.e., v_i, which is an absorbing state. Because the CPS network we consider has only a single source, there is only one absorbing state in the attacker's backtracking process. Consequently, the transition probability of the absorbing state is

$$p_{i,j} = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases} \quad (13)$$

□

***Lemma** 3:* For any given nodes i and j, there exists a time $T_{i,j}(\omega)$ ($T_{i,j}(\omega) < \infty$) that the attacker backtracks to node j for the first time starting from node i (other than the source location). $T_{i,j}(\omega)$, called the first-passage time, is defined as

$$T_{i,j}(\omega) = \min\{n : X_0 = i, X_n(\omega) = j, n \geq 1\} \quad (14)$$

where X_0 is the initial state, and $X_n(\omega)$ is the state at time n from the initial state through the state transition strategy ω.

Proof: According to the definition of a connected graph, any two nodes, other than the source node, are connected, and any other node can reach the source node. Therefore, there must be a time at which backtracking to node j occurs for the first time starting from node i (except the source location). Furthermore, in light of Lemma 1, we can see that the movement of the attacker is a Markov chain. Consequently, according to the definition of the first-passage time in a Markov chain, the attacker's backtracking mode by state transition satisfies Eq. (14). □

***Theorem** 1:* For any given nodes i and j, there is an attacker's first-passage probability from node i to node j, where $i, j \in V$ and node i cannot be the source. The first-passage probability can be expressed as

$$f_{i,j}^{(n)} = P\{T_{i,j} = n | X_0 = i\} \quad (15)$$

Proof: On the basis of Lemma 3, there is a first-passage time ($T_{i,j}(\omega)$) starting from v_i (position of node i) to reach v_j (position of node j), where v_i is not the source location. As indicated by the Markov property of the attacker's movement, the first-passage probability at time n exists and satisfies Eq. (15).

Furthermore, for any nodes i and j ($i, j \in V$), $1 \leq n \leq \infty$, the first-passage probability satisfies

$$p_{i,j}^{(n)} = \sum\nolimits_{l=1}^{n} f_{i,j}^{(l)} p_{j,j}^{(n-l)} \quad (16)$$

where $p_{j,j}^{(0)} = 1$. □

From Theorem 1, we can obtain the average transition time ($\mu_{i,j}$), namely, the conditional mathematical expectation of $T_{i,j}$, which can be expressed as

$$\mu_{i,j} = E\{T_{i,j} | X_0 = i\} = \lim_{T^{\max} \to \infty} \sum\nolimits_{n=1}^{T^{\max}} n f_{i,j}^{(n)} \quad (17)$$

$\mu_{i,j}$ can be used to estimate the expected time at which the attacker first arrives at the source from an initial location. This value employed to determine the safety period of the source location privacy. To maximize the privacy, we set the optimization objective of the average transition time as

$$\max_{[p_{i,j}]^*} \mu_{Start,Asset}([p_{i,j}]) \quad (18)$$

where $\mu_{Start,Asset}$ is the average transition time required for the attacker to arrive at the source (namely, Asset) from an initial node (namely, Start). Then, the fake message scheduling is modeled as an issue of state transition evaluation.

In the fake message scheduling stage, the goal of state transition matrix estimation is to construct communication links between the nodes. As the network scale increases, the computational complexity of the traditional methods for topology construction increases, and there is no guarantee the optimal solution will be found within a bounded time. Moreover, the trade-off among source location privacy, transmission delay and energy consumption makes the optimization objective more complicated. Consequently, a heuristic algorithm is an available and efficient way to achieve the approximately optimal solution. Recently, studies have shown that topology control has achieved considerable progress in terms of energy efficiency and low-delay transmission in networked systems [32], [33]. Considering the topology control as the framework and the maximum average transition time as the optimization objective, we propose a fake message scheduling scheme to achieve a trade-off among privacy level, transmission delay, and energy consumption. The scheme is a probabilistic selection-based distributed fake message scheduling mechanism, in which each spare node calculates the probability of being a fake source in every period with respect to the self and neighboring states and then determines whether to broadcast the fake message.

Let the probability of broadcasting the fake message v_i be denoted as

$$\begin{aligned} p_i^{T(l)} &= \frac{p_i^{T(l)}(\alpha)}{p_i^{T(l)}(\beta)} \qquad (19) \\ &= \frac{\alpha \exp^{\frac{|I_i^B|}{|I_i|}}}{\beta \exp^{1-\frac{rank_{v_i}^E}{|I_i|}} + (1-\beta)\exp^{C(p_i^{T(l-1)})-\frac{|I_i^C|}{|I_i|}}} \end{aligned}$$

where α and β are the weight of a proportion of the backbone set of neighbors and the weight of the transmission delay and energy consumption, respectively, where $\alpha, \beta \in [0,1]$. Both α and β are estimated and discussed in the subsequent sensitivity analysis. I_i^B refers to the neighboring backbone set of v_i, and $rank_{v_i}^E$ denotes the residual energy of v_i in the rank with its neighbors, e.g., $rank_{v_i}^E = 1$ if v_i has more residual energy than the neighboring nodes in the current period. I_i^C represents the set of neighboring nodes of v_i that broadcast the fake message in the previous period. $C(p_i^{T(l-1)})$ denotes a binary function indicating whether v_i broadcasts the fake message in the previous period

$$C(p_i^{T(l-1)}) = \begin{cases} 1 & \text{if } v_i \text{ broadcast at the previous period} \\ 0 & \text{otherwise} \end{cases} \qquad (20)$$

As shown in Eq. (19), several factors influence the probability of broadcasting the fake messages, such as the number of neighboring backbone nodes, the energy consumption of a node and its neighbors, and the status of the fake message broadcasting in the previous period. The greater the number of neighboring backbone nodes is, the higher the probability that a node broadcasts the fake message. By contrast, the lower the energy consumption and the number of fake messages broadcast by itself and its neighbors in the previous period are, the lower the probability. Therefore, a node with a greater number of neighboring backbone nodes or higher residual energy or a node that has not broadcast any fake messages but whose neighbors have broadcast many fake messages in the previous period will be a fake source with a higher probability.

In addition, the bound of $p_i^{T(l)}$ is determined by α and β. We can obtain the bounded values of $p_i^{T(l)}$ in terms of the status of the neighbors as follows:

$$\begin{aligned} p_i^{T(l)} &= \frac{p_i^{T(l)}(\alpha)}{p_i^{T(l)}(\beta)} \in \left(\frac{\alpha}{\exp}, \frac{\alpha \exp}{\beta + (1-\beta)/\exp}\right] \\ &= (\text{Low}, \text{Up}] \qquad (21) \end{aligned}$$

where $p_i^{T(l)} \leq 1$ and $\beta \in [0,1]$, such that Up $\in [\alpha/\exp, \alpha \exp]$. Then, $\alpha \leq 1/\exp$ can be determined. Therefore, $\alpha \in [0, 1/\exp], \beta \in [0,1]$.

Ultimately, we propose a fake message scheduling scheme in which the event in each period contains three main phases. Step 1: each node broadcasts a query for updating the status of the neighboring nodes. Step 2: each node calculates $p_i^{T(l)}$ and then determines whether to become a fake source. Step 3: the source transmits the data along the backbone while the fake sources broadcast fake messages. Then, the process enters the next period and returns to step 1. The pseudocode of the fake message scheduling is given in Algorithm 5.

Algorithm 5 Fake message scheduling

```
Require: A connected network $G$, a backbone connected source and sink, constants $\alpha$, $\beta$.
Queue.push(($sink$, $level = 0$))
for $l = 1$ to $T^{max}$ do
    for $v_i$ in $V$ and $v_i$ is not in backbone do
        broadcast query to neighbors
        update $rank_{v_i}^E$, $I_i^C$, $C(p_i^{T(l-1)})$
        if $l == 1$ then
            calculate $I_i^B$
        end if
        calculate $p_i^{T(l)}$ by using Eq. (19)
        generate a random number $RAND$ within $[0,1)$
        if $RAND < p_i^{T(l)}$ then
            $v_i$ becomes a fake source at period $l$
        end if
    end for
    send data from source to sink along backbone
end for
```

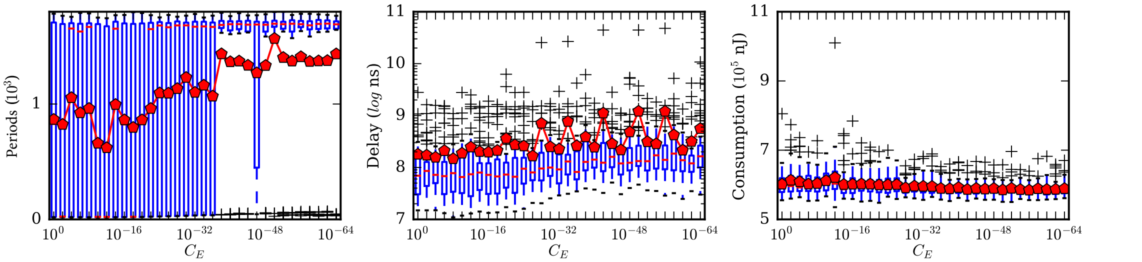
VI. Simulation and performance evaluation

A. Simulation configuration and evaluation metrics

In this section, simulations are conducted to validate the performance of the FSSE algorithm. The specific parameters of the configuration and the compared algorithms are shown in Table III, Table IV, and Table V. First, a sensitivity analysis of FSSE is conducted to obtain the optimal captured probability threshold C_E and to adjust coefficients α and β. Then, the FSSE algorithm is compared with PR [15], TDR [18] and DFSS [22] in terms of source location privacy, transmission delay, and energy consumption. In addition, to eliminate the randomness and occasionality, each group of experiments to verify the performance is performed using 50 connected networks. All simulation results are presented as the average values to ensure the rationality of the experiments.

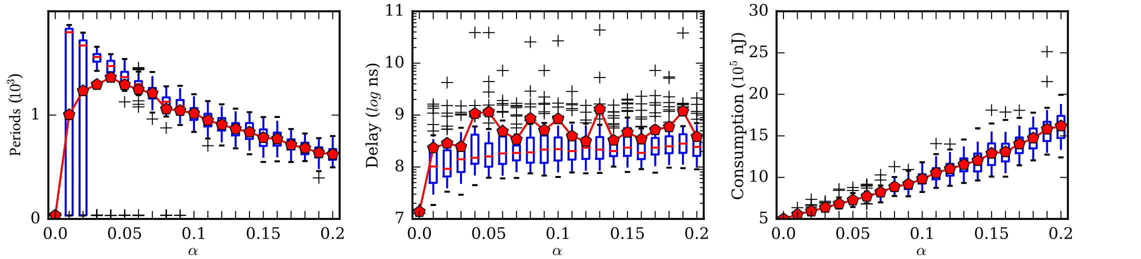
TABLE III
The specific simulation parameters

Parameter	Description	Value
NodeNumber	number of nodes	1000
AreaLength	length of area	1000 m
SinkPosition	location of sink	(333,0)
SourcePosition	location of source	(-333,0)
InitEnergy	initial energy of a node	1 J
Radius	maximum transmitting radius	50 m
InitAttackerPos	initial location of the attacker	(333,0)
T^{max}	maximum period	2000
E^{min}	lower bound of energy	0.01 J



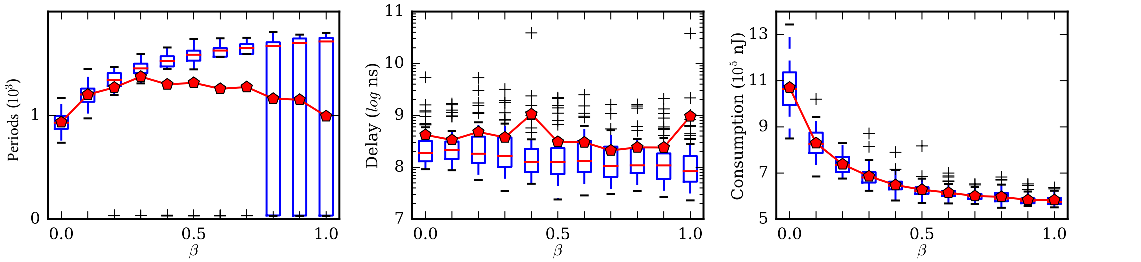
(a) The safety under different C_E (b) The delay under different C_E (c) The energy consumption under different C_E

Fig. 6. Performance analysis under different C_E.



(a) The safety under different α (b) The delay under different α (c) The energy consumption under different α

Fig. 7. Performance analysis under different α.



(a) The safety under different β (b) The delay under different β (c) The energy consumption under different β

Fig. 8. Performance analysis under different β.

TABLE IV
THE MODEL PARAMETERS

Notation	Description	Value
E_{elec}	energy dissipated per bit	50 nJ/bit
ε_{fs}	radio amplifier energy in free space	10 pJ/bit/m^4
ε_{mp}	radio amplifier energy with Rayleigh fading	0.0013 pJ/bit/m^2
E_{da}	energy for data aggregation	5 nJ/bit/signal
d_0	crossover distance	231 m
$t_{i,j}$	packet forwarding capacity	100 ns/bit
p_e	noise power for the environment	0
$g_{i,j}$	parameter of wireless device	9.488×10^{-5}m$^2(d < d_0)$ 5.0625m$^4(d \geq d_0)$
$p^t_{i,j}$	transmitting power	5 nJ/bit/signal

TABLE V
THE PARAMETERS OF THE COMPARED ALGORITHMS

Method	Parameter	Description	Value
Kamat P. [15]	h_{walk}	steps of random walk	10
Long J. [18]	h_{walk}	steps of random walk	10
	Φ	steps to select a relay node	5
	k	number of branches in each ring	0.65
	θ	flag of direction	0.2
	ΔT	duration of each tree	10
Bradbury M. [22]	None	None	None

B. Analysis of the sensitivity parameters

We focus on discussing the sensitive parameters, including the captured probability threshold C_E in backbone construction and the adjustment of coefficients α and β, with respect to the performance of the FSSE algorithm. Assume that these parameters are mutually independent, which can be analyzed by controlling the variables as follows.

1) To study the effect of C_E, with $\alpha = 0.02$ and $\beta = 0.80$, $C_E = \{10^{-i}|i = 0, 2, 4, \ldots, 64\}$.
2) To study the effect of α, with $C_E = 10^{-32}$ and $\beta = 0.80$, $\alpha = \{0.01, 0.02, 0.03, \ldots, 0.20\}$.
3) To study the effect of β, with $C_E = 10^{-32}$ and $\alpha = 0.02$, $\beta = \{0.1, 0.2, 0.3, \ldots, 1.0\}$.

Note that the box plots are drawn for groups of performance scores, which enables us to study the distributed characteristics of a group of scores as well as the levels of the scores. In the box plots, the sorted scores are divided into four equal-sized groups into which 25% of all the scores are placed. The lines dividing the groups are called quartiles, and the groups are referred to as quartile groups. The black crosses represent results far greater than and far less than the average. The red line added to each subfigure indicates the average value of the test performance.

Fig.6 shows the impacts of different C_E values on the privacy level (i.e., safety period), transmission delay, and energy consumption. The safety period shows an increasing trend in Fig.6(a) as the captured probability threshold C_E decreases exponentially. Some fluctuating points are observed at several few safety periods, e.g., a shorter safety period occurs if $C_E = 10^{-10}$ or 10^{-12}, and a longer safety period is obtained if $C_E = 10^{-38}$ or 10^{-50}. As shown in Fig.6(b), the transmission delay shows a minor increase with decreasing C_E because a smaller C_E may lead to a longer backbone, which increases both the number of hops between the source and the sink and the transmission time. Fig.6(c) shows that the change in C_E has almost no influence on energy consumption, which demonstrates that the length of the backbone has little impact on energy consumption when the paths are close to each other. Consequently, the C_E influences mainly privacy and delay: the safety period can increase by a factor of two and the transmission delay can increase by a factor of five as C_E decreases. Finally, we choose $C_E = 10^{-40}$ to achieve a trade-off between the safety period and transmission delay.

In Fig.7, we can see the relationships between α and the privacy level, transmission delay and energy consumption. Fig.7(a) shows an optimal α (namely, α^*, and $\alpha^* \in [0.02, 0.06]$) that maximizes the privacy level exists. Furthermore, Fig.7(b) and Fig.7(c) indicate that both the transmission delay and energy consumption increase as α increases from 0.02 to 0.06. The largest transmission delay is at least as twice the minimum delay, while the largest energy consumption is at most a quarter more than the smallest. Clearly, the impact of α on the transmission delay is greater than that on energy consumption. According to Eq. (19), nodes that are adjacent to the backbone with a large $p_i^{T(l)}$ generate plenty of fake sources, so both the transmission delay of the backbone and the energy consumption decrease. Therefore, we choose $\alpha = 0.02$ to satisfy both the transmission delay and energy consumption.

Fig.8 shows the impact of different β values on the privacy level, transmission delay, and energy consumption. In Fig.8(a), there is an optimal β that achieves the best privacy (namely, β^*, $\beta^* \in [0.2, 0.7]$). The median safety period increases with increasing β, but the privacy become more unstable. Since a large β tends to increase $p_i^{T(l)}(\beta)$ in Eq. (19), it reduces the probability that nodes broadcast fake messages. Therefore, the number of fake sources will decrease, which results in a fake-source hole in the initial execution phase. By contrast, Fig.8(b) and Fig.8(c) show that the FSSE algorithm incurs a small reduction in transmission delay and a large drop in energy consumption, e.g., the energy consumption for $\beta = 0.7$ is 20% less than for $\beta = 0.2$. A small β appears to be better, but the privacy tends to become more unstable. Consequently, we select $\beta = 0.6$ to ensure a stable privacy level and achieve the trade-offs with the other performance metrics.

In accordance with the aforementioned discussion, we choose the optimal values for sensitive parameters, which are summarized in Table VI, as the settings for the comparison experiments with the other algorithms.

TABLE VI
THE OPTIMAL PARAMETERS OF FSSE

Parameter	C_E	α	β
Value	10^{-40}	0.02	0.60

C. Performance Evaluation

The proposed FSSE algorithm is compared with three other algorithms: PR, TDR, and DFSS. In this paper, we focus on the safety period (i.e., privacy level), the transmission delay and energy consumption performance of the four algorithms. The simulation results are obtained from 50 randomly generated networks, which are shown in Fig.9 to Fig.14.

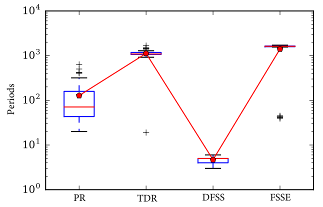


Fig. 9. The safety period under the four algorithms.

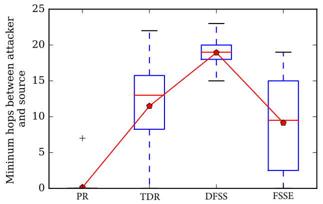


Fig. 10. The minimum number of hops between the attacker and the source.

Fig.9 compares the safety periods of the four algorithms, which verifies the privacy-aware effectiveness. Clearly, the proposed FSSE algorithm has a much longer safety period than PR and DFSS and a slightly longer than TDR. On the other hand, several situations of low safety periods are observed for all four algorithms. Even FSSE has 4 cases with a short safety period because the number of fake sources may be less initially when using a random scheduling strategy while the generated fake sources are simultaneously far from the backbone. As a result, the fake sources adjacent to the backbone cannot efficiently entice the attacker, so the source location is quickly found by the attacker. FSSE still has the best privacy-aware performance among the algorithms. By contrast, DFSS has the lowest privacy-aware capability, mainly due to the uncertainty of the attacker's random walk. Since the fake sources flood an abundance of fake messages, they increase the energy consumption of each node, as well as node failures. Therefore, FSSE achieves a high level of privacy preservation with little additional communication overhead.

Furthermore, we analyze the capability of the four algorithms to confuse the attacker on the basis of the minimum number of hops between the source and the attacker. Usually, the greater the minimum number of hops between the source and the attacker is, the better the capability to confuse the attacker. As shown in Fig.10, PR provides the worst case, which demonstrates that it cannot confuse the attacker efficiently. By contrast, DFSS achieves the best results, effectively keeping the attacker away from the source. On the basis of the above analysis, the safety period in DFSS is short, enabling the attacker to backtrack only for a short duration. Therefore, DFSS would not achieve better privacy-preserving capability even if it has a larger minimum number of hops between the source and the attacker. Both FSSE and TDR protect the source by making the attacker backtrack via certain hops. The proposed FSSE can ensure that the source is not captured by the attacker even when the number of hops between the source and the attacker is less than 5 because we take the previously broadcast fake messages into consideration, which makes the distribution of fake sources approximately uniform. Consequently, the nearly uniform density efficiently entices the attacker to fake sources and ensures a high level of privacy in every period.

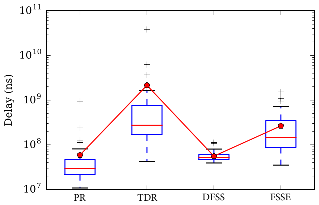


Fig. 11. The transmission delay under different methods.

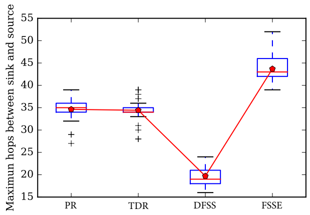


Fig. 12. The maximum number of hops between the sink and the source.

Fig.11 and Fig.12 show the transmission delay performance, i.e., the latency and the number of hops from the source to the sink, of the four algorithms. As shown in Fig.11, both PR and DFSS provide lower latency than FSSE. Meanwhile, TDR has the longest latency. In Fig.12, DFSS has the shortest backbone, and the FSSE has the longest backbone. DFSS constructs the shortest path between the source and the sink by implementing flooding routing. PR and TDR introduce a random walk for selecting a phantom node, which increases the number of hops. Usually, a longer backbone improves privacy. FSSE constructs the longest backbone, but only a small number of fake sources are used for enticing the attacker. Therefore, FSSE uses fake message scheduling to avoid the additional transmission delay caused by the longer backbone, which results in a delay far less than that of DFSS.

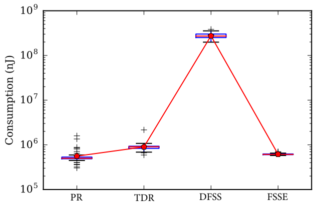


Fig. 13. The energy consumption under different methods.

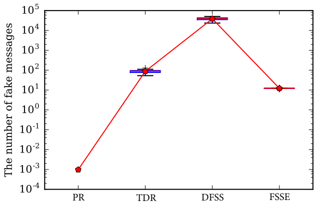


Fig. 14. The average number of fake messages broadcast in each period.

Fig.13 and Fig.14 show the energy consumption, i.e., the network lifetime and the number of fake messages broadcast, under the four algorithms. As shown in Fig.13, the energy consumption of FSSE is second only to PR. However, FSSE provides more stable energy dissipation compared with the overhead in PR. A probability-based selection is employed to schedule fake sources, that is, no node is always broadcasting fake messages. Thus, scheduling idle nodes only slightly increases energy consumption; nevertheless, the main energy consumption is a result of transmitting the data on the backbone.

Fig.14 shows a positive correlation between the number of fake messages and the communication overhead. Because no fake messages are broadcast, PR has the minimum communication overhead, but its privacy level is unstable. Compared with TDR and DFSS, FSSE broadcasts the fewest fake messages and achieves the lowest communication cost. The average number of fake messages broadcast is less than 13, which indicates that the proportion of fake sources in the network is less than 1.3%. Consequently, we also find that the number of fake sources near the backbone is too small to result in substantially more overhead on the backbone.

VII. Conclusion and Future Work

In this paper, we address the problem of scheduling fake sources to enhance source location privacy and maintain system performance. FSSE is proposed to defend the sources of cyber-physical systems against a random-walking attacker. The proposed algorithm contains two main phases. First, backbone construction is presented considering the likelihood of capturing the source. Fake message scheduling then is established based on the hypothesized location of the attacker, which is evaluated by using stochastic processes. The simulation results show that FSSE efficiently defends against the attacker and has a more stable privacy level and a more efficient system performance (with respect to transmission delay and energy consumption) than the compared algorithms: PR, TDR, and DFSS.

However, the limitations of this study are as follows. (1) The proposed algorithm is only built for a wireless network environment with sensors and actuators. (2) The effectiveness of the proposed algorithm is verified in simulation, but we do not prove the efficiency in the real physical world, such as the actual environment with full of the interference and background noise. (3) The efficiency of the proposed algorithm under the attacker model with active learning capability is not further proved. Therefore, in view of the difference between simulation and real-world applications, we make a follow-up future work: (1) Building small and medium-sized process industrial control systems that are close to smart factories, such as the testbed of the water-level control system. (2) Studying the hybrid mode of the possible network and physical attacks, and build an attacker model based on active learning. (3) Establishing a smart offensive and defensive game model for complex systems.

References

[1] W. Wolf, “Cyber-physical systems,” *Computer*, vol. 42, no. 3, pp. 88–89, 2009.

[2] R. Baheti and H. Gill, "Cyber-physical systems," *The impact of control technology*, vol. 12, pp. 161–166, 2011.

[3] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao, "A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2551–2566, 2017.

[4] Y. Zhang, M. Qiu, C. W. Tsai, M. M. Hassan, and A. Alamri, "Health-cps: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Systems Journal*, vol. 11, no. 1, pp. 88–95, 2017.

[5] W. Dai, V. N. Dubinin, J. H. Christensen, V. Vyatkin, and X. Guan, "Toward self-manageable and adaptive industrial cyber-physical systems with knowledge-driven autonomic service management," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 725–736, 2017.

[6] R. Pal and V. Prasanna, "The stream mechanism for cps security the case of the smart grid," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 537–550, 2017.

[7] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, 2011.

[8] A. Nourian and S. Madnick, "A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 2–13, 2018.

[9] W. Zhang, *Source Location Privacy*. Boston, MA: Springer US, 2011, pp. 1230–1231.

[10] P. Derler, E. A. Lee, and A. L. Sangiovanni-Vincentelli, "Addressing modeling challenges in cyber-physical systems," DTIC Document, Report, 2011.

[11] J. Chen, Z. Lin, Y. Hu, and B. Wang, "Hiding the source based on limited flooding for sensor networks," *Sensors*, vol. 15, no. 11, p. 29129, 2015.

[12] M. Conti, J. Willemsen, and B. Crispo, "Providing source location privacy in wireless sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1238–1280, 2013.

[13] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proceedings of the 2Nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*. New York, NY, USA: ACM, 2004, pp. 88–93.

[14] K. Mehta, D. G. Liu, and M. Wright, "Protecting location privacy in sensor networks against a global eavesdropper," *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 320–336, 2012.

[15] P. Kamat, Z. Yanyong, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, 2005, pp. 599–608.

[16] Y. Li and J. A. Ren, "Source-location privacy through dynamic routing in wireless sensor networks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–9.

[17] P. Kumar, J. P. Singh, P. Vishnoi, and M. P. Singh, "Source location privacy using multiple-phantom nodes in wsn," in *TENCON 2015 - 2015 IEEE Region 10 Conference*, 2015, pp. 1–6.

[18] J. Long, M. X. Dong, K. R. Ota, and A. F. Liu, "Achieving source location privacy and network lifetime maximization through tree-based diversionary routing in wireless sensor networks," *IEEE Access*, vol. 2, pp. 633–651, 2014.

[19] A. Jhumka, M. Bradbury, and M. Leeke, "Towards understanding source location privacy in wireless sensor networks through fake sources," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, 2012, pp. 760–768.

[20] A. Thomason, M. Leeke, M. Bradbury, and A. Jhumka, "Evaluating the impact of broadcast rates and collisions on fake source protocols for source location privacy," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013, pp. 667–674.

[21] A. Jhumka, M. Bradbury, and M. Leeke, "Fake sourcebased source location privacy in wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 12, pp. 2999–3020, 2015.

[22] M. Bradbury, M. Leeke, and A. Jhumka, "A dynamic fake source algorithm for source location privacy in wireless sensor networks," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, 2015, pp. 531–538.

[23] J. Ren, Y. Li, and T. Li, "Routing-based source-location privacy in wireless sensor networks," in *2009 IEEE International Conference on Communications*, 2009, pp. 1–5.

[24] R. H. Hu, X. M. Dong, and D. L. Wang, "Protecting data source location privacy in wireless sensor networks against a global eavesdropper," *International Journal of Distributed Sensor Networks*, p. 17, 2014.

[25] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal denial-of-service attack scheduling with energy constraint," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 3023–3028, 2015.

[26] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *2009 IEEE INFOCOM*, 2009, pp. 2213–2221.

[27] Y. Fan, J. Chen, X. Lin, and X. Shen, "Preventing traffic explosion and achieving source unobservability in multi-hop wireless networks using network coding," in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, 2010, pp. 1–5.

[28] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Workshop on future directions in cyber-physical systems security*, vol. 5, 2009.

[29] J. S. Gui and Z. W. Zeng, "Joint network lifetime and delay optimization for topology control in heterogeneous wireless multi-hop networks," *Computer Communications*, vol. 59, pp. 24–36, 2015.

[30] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

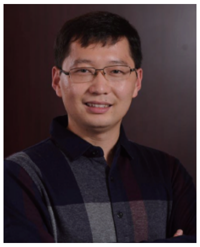
[31] R. Shi, M. Goswami, J. Gao, and X. F. Gu, "Is random walk truly memoryless - traffic analysis and source location privacy under random walks," in *2013 Proceedings IEEE INFOCOM*, 2013, pp. 3021–3029.

[32] J. He, S. L. Ji, R. Beyah, Y. Xie, and Y. S. Li, "Constructing load-balanced virtual backbones in probabilistic wireless sensor networks via multi-objective genetic algorithm," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 2, pp. 147–163, 2015.

[33] Z. Hong, R. Wang, and X. Li, "A clustering-tree topology control based on the energy forecast for heterogeneous wireless sensor networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 1, pp. 68–77, 2016.

Zhen Hong received a Bachelor of Computer Science & Technology and Computing from Zhejiang University of Technology (China) and University of Tasmania (Australia) in 2016, respectively, and a Ph.D. from the Zhejiang University of Technology in 2012. He was an associate professor with the Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, China. Now he is an associate professor with the Institute of Cyberspace Security, Zhejiang University of Technology, China. Dr. Hong has visited at the Sensorweb Lab, Department of Computer Science, Georgia State University for 3 months in 2011. Since Sept. 2016, he has been at CAP Research Group, School of Electrical & Computer Engineering, Georgia Institute of Technology as a research scholar. His research interests include cyber-physical systems, Internet of things, wireless sensor networks, cybersecurity, and data analytics. He received the first Zhejiang Provincial Young Scientists Title in 2013 and the Zhejiang Provincial New Century 151 Talent Project (The Third-Level) in 2014. He also received the 521 Talent Project of Zhejiang Sci-Tech University and the Young and Middle-aged Talents Foundation of Zhejiang Provincial Top Key Academic Discipline of Mechanical Engineering in 2015. He has published more than 30 papers and applied for more than 60 Chinese patents (25 patents have been authorized). He is a member of CCF and CAA, and serves on the Youth Committee of Chinese Association of Automation.

Rui Wang is a master student of Measurement Technology and Instruments from the Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, China. His main research interests include cyber-physical systems, wireless sensor networks, cybersecurity and data mining. He received First Prize and the HUAWEI Special Award of the China Post-Graduate Mathematical Contest in Modeling in 2016.



Shouling Ji is a ZJU 100-Young Professor in the College of Computer Science and Technology at Zhejiang University and a Research Faculty in the School of Electrical and Computer Engineering at Georgia Institute of Technology. He received a Ph.D. in Electrical and Computer Engineering from Georgia Institute of Technology and a Ph.D. in Computer Science from Georgia State University. His current research interests include big data security and privacy, big data driven security and privacy, and adversarial learning. He is a member of IEEE and ACM and was the Membership Chair of the IEEE Student Branch at Georgia State (2012-2013).

Raheem Beyah is the Motorola Foundation Professor and Associate Chair for Strategic Initiatives and Innovation in the School of Electrical and Computer Engineering at Georgia Tech, where he leads the Communications Assurance and Performance Group (CAP) and is affiliated with the Institute for Information Security & Privacy (IISP). He received his Bachelor of Science in Electrical Engineering from North Carolina A&T State University in 1998. He received his Masters and Ph.D. in Electrical and Computer Engineering from Georgia Tech in 1999 and 2003, respectively. Dr. Beyah has served as a Guest Editor for MONET and is currently an Associate Editor of the (Wiley) Wireless Communications and Mobile Computing Journal. His research interests include network security, wireless networks, network traffic characterization and performance, and critical infrastructure security. He received the National Science Foundation CAREER award in 2009 and was selected for DARPA's Computer Science Study Panel in 2010.