Computers, Materials & Continua
DOI: 10.32604/cmc.2022.026657
Article

Tech Science Press

Network Traffic Obfuscation System for IIoT-Cloud Control Systems

Yangjae Lee[1], Sung Hoon Baek[2], Jung Taek Seo[3] and Ki-Woong Park[1,*]

[1]SysCore Lab., Sejong University, Seoul, 05006, Korea
[2]Department of Computer System Engineering, Jungwon University, Chungcheongbuk-do, 28024, Korea
[3]Department of Computer Engineering, Gachon University, Gyeonggi-do, 13120, Korea
*Corresponding Author: Ki-Woong Park. Email: woongbak@sejong.ac.kr
Received: 31 December 2021; Accepted: 11 March 2022

Abstract: One of the latest technologies enabling remote control, operational efficiency upgrades, and real-time big-data monitoring in an industrial control system (ICS) is the IIoT-Cloud ICS, which integrates the Industrial Internet of Things (IIoT) and the cloud into the ICS. Although an ICS benefits from the application of IIoT and the cloud in terms of cost reduction, efficiency improvement, and real-time monitoring, the application of this technology to an ICS poses an unprecedented security risk by exposing its terminal devices to the outside world. An adversary can collect information regarding senders, recipients, and prime-time slots through traffic analysis and use it as a linchpin for the next attack, posing a potential threat to the ICS. To address this problem, we designed a network traffic obfuscation system (NTOS) for the IIoT-Cloud ICS, based on the requirements derived from the ICS characteristics and limitations of existing NTOS models. As a strategy to solve this problem wherein a decrease in the traffic volume facilitates traffic analysis or reduces the packet transmission speed, we proposed an NTOS based on packet scrambling, wherein a packet is split into multiple pieces before transmission, thus obfuscating network analysis. To minimize the ICS modification and downtime, the proposed NTOS was designed using an agent-based model. In addition, for the ICS network traffic analyzer to operate normally in an environment wherein the NTOS is applied, a rule-based NTOS was adopted such that the actual traffic flow is known only to the device that is aware of the rule and is blocked for attackers. The experimental results verified that the same time requested for response and level of difficulty of analysis were maintained by the application of an NTOS based on packet scrambling, even when the number of requests received by the server per second was reduced. The network traffic analyzer of the ICS can capture the packet flow by using the pre-communicated NTOS rule. In addition, by designing an NTOS using an agent-based model, the impact on the ICS was minimized such that the system could be applied with short downtime.

Keywords: Cloud computing system; moving-target defense; container orchestration

1 Introduction

Efforts continue to enhance the performance of industrial control systems (ICSs) for remote control, operational efficiency improvement, real-time big-data monitoring, and incident response. One of the latest core-based technologies that enable the achievement of this goal is the ICS, which comprises an integration of the Industrial Internet of Things (IIoT) and the cloud [1,2]. This IIoT-Cloud ICS can achieve the objectives of cost reduction, efficient operation, real-time monitoring, and responses by managing industrial machines in a remote cloud rich in resources. Industrial machines can be connected to the Internet directly or through a remote terminal unit (RTU) that manages and controls them according to their respective configurations [3]. However, with the external exposure of these terminal machines, the IIoT-Cloud ICS poses a security risk, unlike conventional ICSs [4].

In a conventional ICS with integrated networks, all industrial machines are connected to the Internet through the ICS network [5–9]. Even if attackers can determine the traffic flow by analyzing the packet size, inter-packet timing, and headers in their traffic analysis attacks [10], they can only view the integrated packets transmitted through the networks involved, and not the details of industrial machines, users, and time slots. In contrast, as depicted in Fig. 1a, if industrial machines are directly connected to the Internet, as are the IIoT-Cloud ICS, traffic analysis attackers can view the details of industrial machines, the senders and recipients of the packets, and their prime-time slots [11]. To address this problem, an IIoT-Cloud ICS should be equipped with a security measure to prevent network analysis to protect the ICS against potential threats arising from exposure. Network security researchers have conducted intensive research to prevent traffic analysis, and one of the techniques proposed is the network traffic obfuscation system (NTOS). To hinder traffic analysis, this system uses strategies, such as the uniformization of the packet size [10], concealment of the header's source and destination IP addresses [11], and anti-analysis of inter-packet timing through packet delay in the proxy server, to make it difficult to identify the packet flow [12].

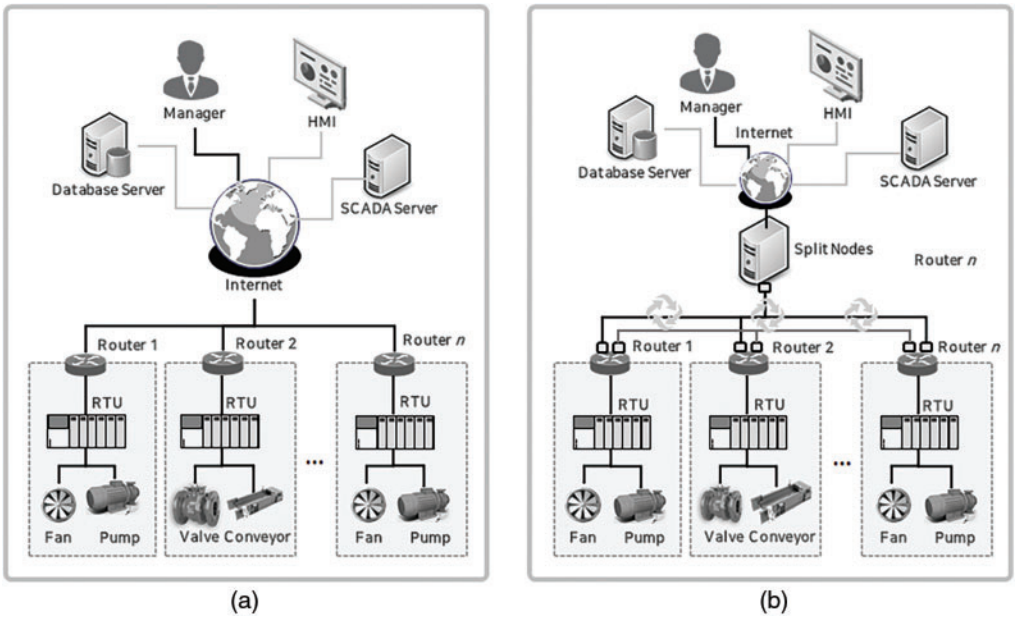

Figure 1: (a) Conventional network architecture of the IIoT-Cloud Industrial Control System, (b) IIoT-Cloud Industrial Control System enhanced with a network traffic obfuscation technology

However, conventional NTOSs have the following limitations: First, their application reduces the speed and the level of analysis difficulty, which can pose a fatal risk to the ICS with an irregular and relatively small traffic volume in comparison with that of general networks [13]. In addition, conventional systems lose speed at each mix node by running encryption/decryption operations [14]. Some systems, such as Tor [15], are vulnerable to passive traffic analysis attackers capable of viewing all external networks [16]. To defeat such attackers, an NTOS that is unaffected by a decrease in the traffic volume must be developed. However, it is a great challenge to apply the NTOS to an ICS, because the latter contains expensive machines that cannot be modified [17] easily, and the system downtime can cause enormous losses, which must be minimized when applying a new security measure. Therefore, when applying an NTOS, the modification and downtime in the ICS components and network should be reduced to the minimum. In addition, an ICS runs a network traffic analyzer for security reasons [18]. Because the analyzer is turned off when the traffic obfuscation system is being applied, a measure should be taken to ensure the normal operation of the analyzer.

This study proposes an NTOS based on packet scrambling to prevent the network analysis of an IIoT-Cloud ICS, as illustrated in Fig. 1b, by extracting the specifications from the characteristics of the ICS and the limitations of the NTOS. Unlike conventional NTOSs, wherein a decrease in the traffic volume is associated with reduced speed or difficulty, an NTOS based on packet scrambling is characterized by the maintenance of the analysis difficulty level and packet response time by sending the segmented packets to multiple agents. To prevent the attacker from tracing back to the sender following a response from any of the agents, the time required for the response of the non-receiving agents is also communicated in advance. To deceive the attacker, non-receiving agents delay the transmission of dummy packets for the response time required by the receiving agents, and then transmit the dummy packets to the split server. The rule-based NTOS generates packets of a uniform size according to the pre-communicated rule when segmenting a packet to ensure the normal operation of the ICS network traffic analyzer. The network traffic analyzer uses the rule to identify the destination of the packet, which is impossible for an attacker who is unaware of the rule. In addition, the NTOS was designed using an agent-based model to enable, with the minimum necessary modifications and downtime, the application of the NTOS to the ICS. This design enables the application of the NTOS quickly with the least possible modifications to the existing ICS components.

The contributions of this paper are threefold. First, the use of the packet scrambling-based NTOS enables the traffic speed and analysis difficulty level to be maintained even when the traffic volume of the industrial machines decreases to prevent global passive network analysis attackers. Second, the use of the rule-based NTOS enables the ICS network traffic analyzer to continue operating normally even during the NTOS application because only the network traffic analyzer, being aware of the rule, can identify the actual traffic flow. Third, the agent-based communication among the ICS components minimizes their modifications and system downtime. The remainder of this paper is organized as follows: Section 2 investigates the ICS characteristics and identifies the limitations of the NTOS: Section 3 defines the requirements for the design based on the ICS characteristics and the limitations of the NTOS identified in Section 2, and describes the design and architecture of the packet scrambling-based NTOS, reflecting the defined requirements. Section 4 presents the experiments and verification of the requirements, and Section 5 draws the conclusions.

2 Literature Review & Related Work

2.1 Traffic Analysis Attack

A traffic analysis attacker seeks to obtain significant information through packet monitoring and metadata analysis, as well as bypassing the packet encryption key in conjunction with the tunneling protocol [19]. A network traffic analysis attacker collects a large number of packets on networks with security applications, such as virtual private networks (VPNs) and encryption, and analyzes metadata, the unencrypted portions of a packet, such as the packet header, packet size, and inter-packet timing. The attacker deduces the IP addresses of the source and destination from the packet header and identifies the packet flow based on the packet size. In addition, when a packet passes through the VPN server, it is possible to connect inbound and outbound packets by determining their interrelationships, such as inter-packet timing and order [20]. Based on point-to-point packet flow, the attacker can deduce information regarding the sender and receiver, and regarding the communication entities, communication frequency, and prime traffic time of specific users [21]. Alone, such information is not fatal but can be linked to other attacks or used for subsequent attacks. For example, an attacker who can identify a network user by analyzing the network traffic can selectively block a specific user's communication by launching a distributed denial-of-service attack on the IP address during communication. Alternatively, IP information intercepted by users in frequent communication with the ICS can be used as the linchpin for penetrating vulnerable ICS devices.

2.2 ICS Characteristics

ICS is a generic term used to refer to the control systems used by industrial entities as well as national infrastructures, such as power plants, banks, and hospitals, to control their equipment. An ICS has many characteristics different from those of general IT systems because the risks that arise in it can pose threats to safety, the social order, and economic stability [17]. Therefore, the application of security measures of the ICS requires particular attention. First, modification to the ICS components and system downtime should be minimized when the security measures are applied. ICSs are interdependent because of their close interconnectedness [17], and downtime in one ICS significantly undermines the entire network. For example, downtime in a nuclear power plant costs approximately GBP 33,000 per hour [22]. In addition, given that an ICS contains expensive machines that are difficult to replace [18], a lengthy system downtime while applying modification to it is highly problematic. Second, an ICS should be able to defeat a global passive adversary. A global passive adversary is an attacker capable of eavesdropping anywhere on external networks [16] and is also capable of tracing packet flow even if an NTOS such as Tor is applied by determining the relational flow of traffic. Owing to the nature of ICS' risks being directly linked to real-world threats, an ICS is a major target for cyberwarfare attackers [23] or terrorists [24] seeking to cause social disruption. Because these adversaries can wage attacks based on the same capabilities as global passive adversaries (GPAs), backed by government support, their traffic analysis attacks must be prevented by all means. Third, the operation of the network traffic analyzer must be assured and the ICS must detect security events and incidents [17]. For security reasons, many ICSs operate a network traffic analyzer to measure traffic volumes, manage performance, and detect malicious users [18]. However, the NTOS application may lead to downtime or an irregular operation of the network traffic analyzer. Fourth, as the ICS is only accessible to authorized managers and is prohibited for general use, the ICS network traffic volume is small and irregular in comparison with that of general systems, tending to increase or decrease depending on the state of the industrial machines. In certain circumstances, a decrease in traffic volume may lead to a decrease in the performance of the NTOS.

2.3 Chaum's Mixes

The concept of a mix node for anonymous correspondences was first proposed by Chaum in 1981 [12]. In a system with a mix node application, all packets between the user and the destination share mix nodes. In Chaum's mix node model, packets are reshuffled once the threshold is reached, and leave for the next mix node or destination node. In this threshold-based mix node, the order of packets is shuffled such that the mix node renders the targets of the traffic analysis attacks, such as headers and inter-packet timing, which are robust against attacks. However, Chaum's threshold-based NTOS has a limitation because packets become oversized due to excessive asymmetric encryption/decryption operations as they pass through multiple mix nodes, thus increasing the delay per node. More importantly, because packets leave each node only when the threshold number of packets is reached, transmission can be delayed indefinitely [14].

2.4 Stop-and-Go-MIXes

Stop-and-go-MIXes [25] hold back each packet for a random period before transmitting it from each mix node. As each packet is delayed randomly before being further transmitted, the order of entering and leaving the mix node is different. This makes it difficult for an attacker to infer the correlation between the messages that enter and leave the mix node. This type is called a timing-based NTOS. A stop-and-go-MIX NTOS overcomes the limitation of the threshold-based NTOS, wherein a decrease in traffic volume can lead to an infinite delay of transmission, given that the threshold number of packets must be reached before packets can leave each node. However, stop-and-go-MIXes are limited because a decrease in the traffic volume decreases the analysis difficulty level [26] and the random delay designed to maintain the analysis difficulty level can reduce the transmission speed of the entire packet.

2.5 Babel's Mixes

Babel's Mixes [26] combines threshold-based and timing-based NTOS types in which packets are transmitted to the next node after the lapse of a predetermined time even when the threshold number of packets is not reached. In addition, when the traffic volume decreases, dummy messages are transmitted to prevent excessive delays in the transmission of packets. Designed to combine the threshold-based and timing-based NTOS types, Babel's Mixes benefit from the advantages of both models and address their common limitations associated with decreased traffic volume by transmitting dummy packets at regular intervals. However, to maintain the difficulty level of the traffic analysis, it is necessary to increase the similarity to real packets by transmitting dummy packets with patterns similar to real packets. The problem with this process is that, as the design of the dummy packet is not described in detail, there are no clear criteria for assessing the similarity of dummy packets. Another limitation of Babel's Mixes is that the delay in transmission increases as the packets pass through the mix nodes owing to excessive encryption/decryption operations.

2.6 Tor

Tor [15] is an onion-routing-based NTOS designed for fast communication, such as web browsing. Tor's onion-routing network contains a set of multiple nodes called an onion router (OR). The user selects the OR in a predetermined order to create a circuit as a two-way communication channel for data exchange. The user encrypts the packet with the symmetric key of each OR, layer-by-layer. Each OR decodes the data of the pertinent layer and transmits the packet to the next OR. Tor supports fast communication using a queuing strategy such as first-in-first-out. However, because it neither

maintains the analysis difficulty level nor shuffles the packets through packet delay as a trade-off for speed Tor is vulnerable to a global passive adversary that can view all packets [16].

3 Design and Implementation of a Network Traffic Obfuscation System

This section explains the packet scrambling-based NTOS proposed in this study, the delay-based attacker deception technique, the mechanism of the rule-based NTOS, and the flow of the system. The proposed NTOS is composed of managers that access industrial machines remotely, manager agents that support their manager to participate in the NTOS, split nodes that split the incoming packets and transmit the split packets to the group members of the receiving end, industrial machines (machines) under operation, industrial machine agents that support their industrial machines to participate in the NTOS, and the ICS network traffic analyzer. An RTU may or may not be used depending on the architecture of the IIoT-Cloud ICS, and the term "industrial machine" as used in this paper includes the scope of RTUs and refers either to the industrial machine itself or the RTU depending on the architecture of the IIoT-Cloud ICS. RTU controls local industrial machines and plays the role of transmitting incoming requests to industrial machines. In the packet scrambling-based NTOS, manager agents and machine agents are grouped into n groups, as shown in Fig. 2, where the manager agents are not subject to control and have their private keys as a security measure against possible insider attacks. On the contrary, machine agents are assumed to share private keys within the same group because they are subject to control by the ICS equipment manager. It is assumed that there is a means of allowing all components of the proposed NTOS to safely share a public key, IP, and rules of rule-based NTOS. It is also assumed that the addition of new managers or machines is controlled by strict rules other than the protocol.

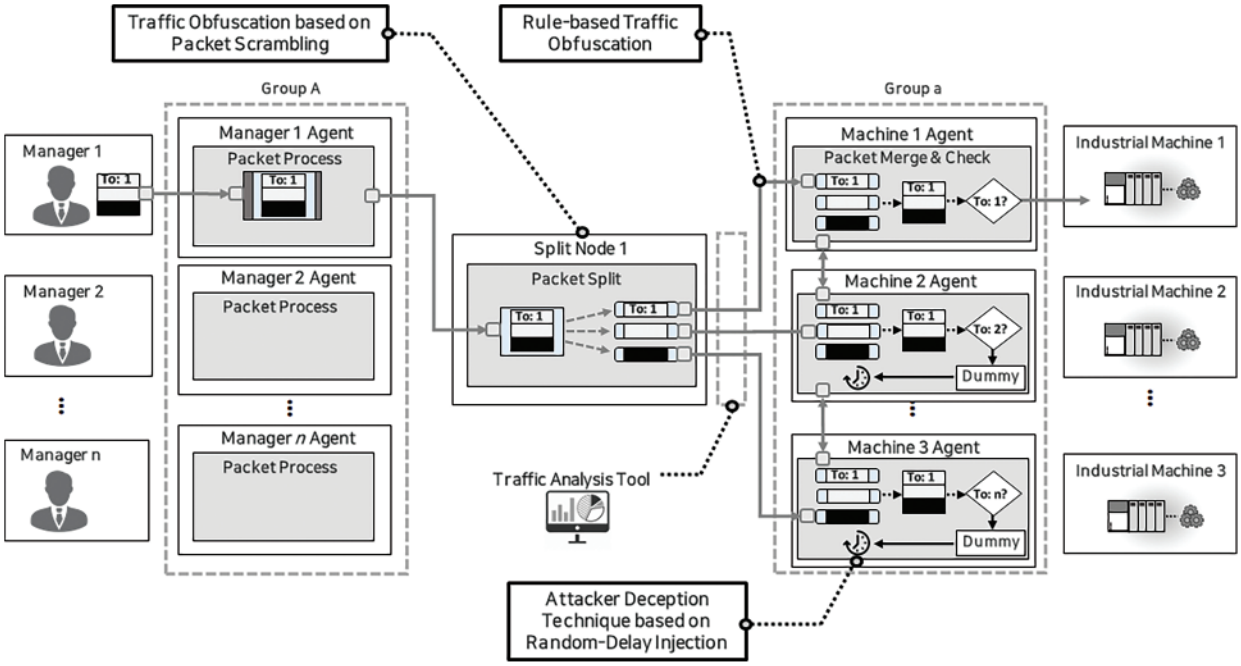

Figure 2: Conceptual overview of traffic obfuscate technique for industrial control system

3.1 Requirements and Attacker Mode

Since problems in an ICS can cause social disruption, human casualties, or fatal harm to the organization, and its characteristics are different compared to those of general systems, its design should meet the following requirement-set based on the ICS characteristics and the limitations of the NTOS:

First, given that ICSs are closely interrelated and contain expensive components that are difficult to replace, modification to ICS components and system downtime due to the NTOS' application should be minimized. To this end, this paper proposes an agent-based NTOS. Second, since ICS problems can directly lead to social disruption or national harm, an ICS is a major target for cyber-warfare attackers or terrorists backed by their governments. Therefore, an NTOS must be able to defeat a global passive adversary that can monitor all external networks. This paper proposes this type of NTOS. Third, the ICS operates a network traffic analyzer to measure the traffic volume, assess performance, and identify global passive adversaries. The operation of the network traffic analyzer may, however, be affected by the application of an NTOS. To ensure the normal operation of the ICS network even during the NTOS' application, this study proposes a rule-based NTOS where the traffic flow can be identified only by the network traffic analyzer that knows the packet splitting rule. Fourth, the ICS network traffic volume is small and irregular in comparison with those of general systems because of the lower number of users. In the conventional types of NTOS, a decrease in the traffic volume leads to a decrease in the analysis difficulty level or an increase in delay. To address these limitations, this paper proposes a packet scrambling-based NTOS that maintains the packet transmission speed and the traffic analysis difficulty level even when the traffic volume decreases.

3.2 Architecture of the Packet Scrambling-based Network Traffic Obfuscation System

This section describes the packet scrambling-based NTOS in which packet splitting and merging strategies are used to defeat global passive adversaries by overcoming the drawback of reduced analytical difficulty and packet transmission speed caused by a decrease in traffic volume. In the packet scrambling-based NTOS, packets generated by a manager are processed by an agent and transmitted to a split node. A split node splits the incoming packet into n packets, in which split messages (to be merged again on the receiving end) are inserted. The packet size is set according to the rule of the rule-based NTOS, and the split packets are transmitted to the agents of the receiving end. The agents share the received split packets within the group to create the original packet by merging them. This split and merge method makes it difficult for the attacker to identify the actual recipient because the original packet is split and transmitted to multiple agents. In addition, unlike the conventional types of NTOS, where the analysis difficulty level is maintained by mixing it with other packets, the analysis difficulty level is as high as the packet itself, which prevents a decrease in the packet transmission speed and analysis difficulty level, even if the traffic volume decreases. A detailed description of the packet splitting and merging processes is given below.

3.2.1 Splitting Process

The split method is designed to hide the recipient from attackers by splitting the traffic in the broadband network and transmitting the split packets to multiple agents on the receiving end, where they are merged. To apply the split method, the packet received by the agent must be preprocessed into a packet for use in the packet scrambling-based NTOS, as shown in Fig. 3a. On receiving the packet from a manager or industrial machine, an agent extracts the data on the source IP and packet and encrypts them with the symmetric key of the destination agent that can be obtained from messages 1 to 3 in State 1. Then, the encrypted packet, destination IP, and padding to set the packet size are encrypted

with the symmetric key of the split node. The agent changes the source IP of the generated packet header to a random value and transmits it to the split node. Upon receiving the encrypted packet from the agent, the split node splits the packet, as shown in Fig. 3b. The split node decrypts the packet with the private key to obtain the destination IP and encrypted data in the packet. In this process, the split node does have information on the additional encrypted packets except for the destination. The split node encrypts the decrypted data with the symmetric key of the destination agents and splits the packet into n packets, matching the number of the group members of the destination IP retrieved from the split node database. The packet is split by word unit, and the split node inserts the packet identification and sequence numbers into the packet for them to be merged again. With padding inserted to a size set by the rule of the rule-based NTOS, the split packets are transmitted to all agents of the destination group.

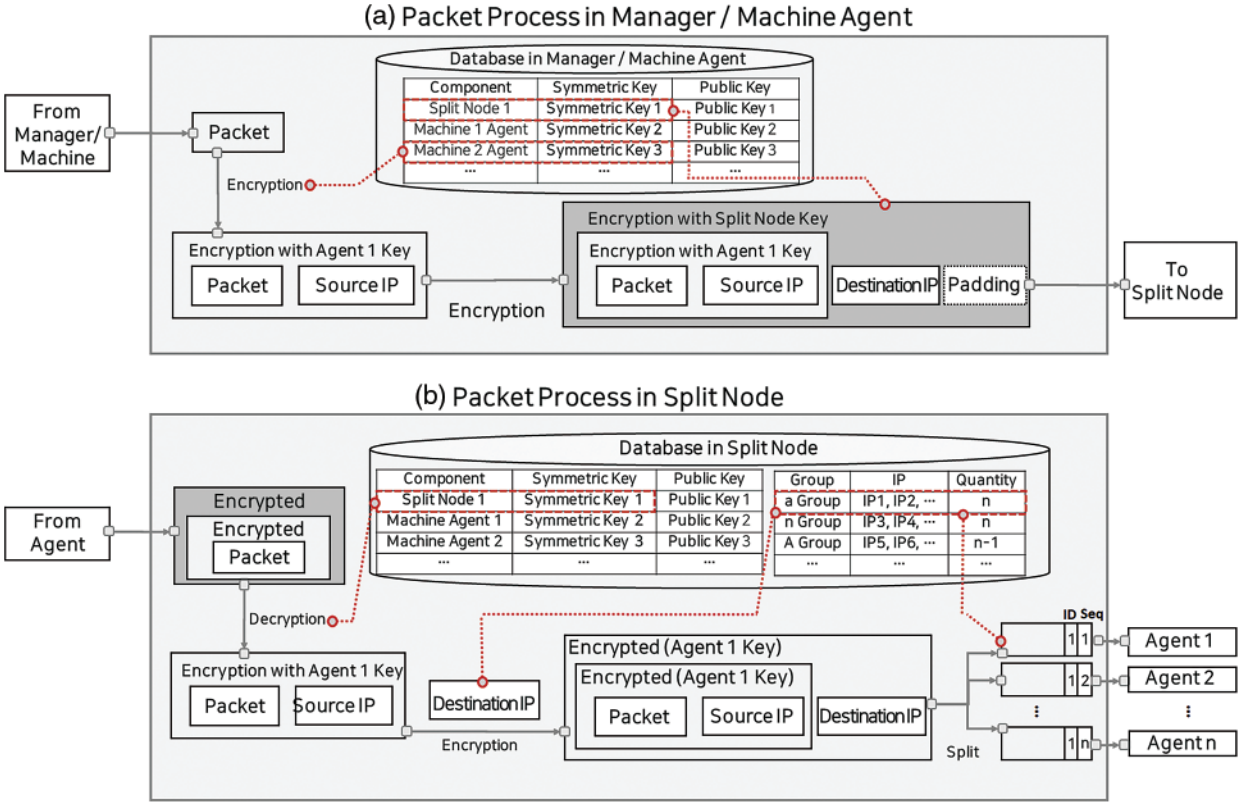

Figure 3: Overall operation flow for packet splitting (a) inside an agent (b) inside a split node

3.2.2 Merging Process

In the merging process, the agent creates the original packet by merging all the split packets it has received from the split node and other agents based on the identification and sequence numbers and checks the destination IP, as shown in Fig. 4. If any of the split packets are missing, the agent requests other agents to send packets. When an industrial machine agent receives the packet, it decrypts it using the group's symmetric key. If a manager agent receives the packet, it again uses its private symmetric key for decryption. The agents on both the sending and receiving ends can retrieve the data in the original packet and the source and destination IP addresses. If the destination IP is identical to the

manager's or to the machine assigned to an agent's IP, the agent transmits the packet and source IP to the manager. Otherwise, the agent analyzes the packet's request, encrypts it with the symmetric key of the split node, and transmits the encrypted dummy packet to the split node after delaying it for the period required as shared among the group.

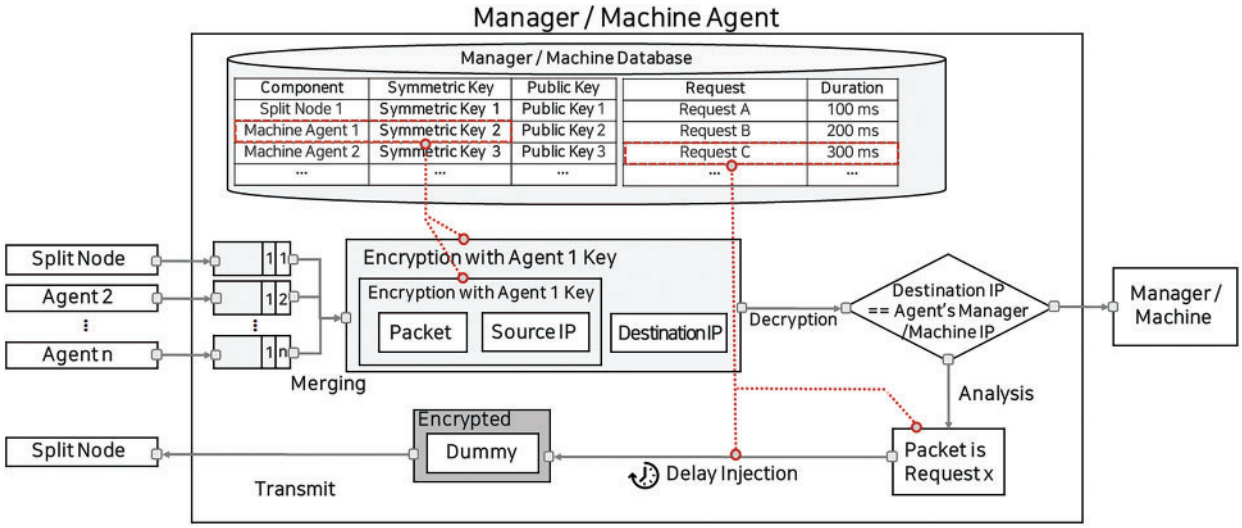

Figure 4: Overall operation flow for packet merging, checking, and deception inside the agent

3.3 Architecture of the Delay-Based Attacker Deception Technique

This section describes the attacker deception technique performed by the agents of the machine agent group to hide the recipient of the original packet in the response process. It is assumed here that the response time required for the inter-agent requests is shared among them. Fig. 5 illustrates an example of the attacker's deception technique. Machine agents merge the split packets to create the original packet and the destination IP. If the destination IP is identical to its machine IP, the machine agent transmits the packet to that machine. In the example implementation in Fig. 5, it takes 230 ms for Machine 1 to receive the packet and generate the response. However, if the destination IP is different from the machine agent's IP, it generates a dummy packet, encrypts it with the private key of the split node, and responds to the split node after delaying the dummy packet for the time required for carrying out the corresponding request earlier. Since it took 230 ms for Machine 1 to receive the packet and generate the response (Fig. 5), the other agents transmit the dummy packet to the split node after delaying it for 229∼231 ms. A response is sent concurrently so that the attacker does not know which industrial device is communicating.

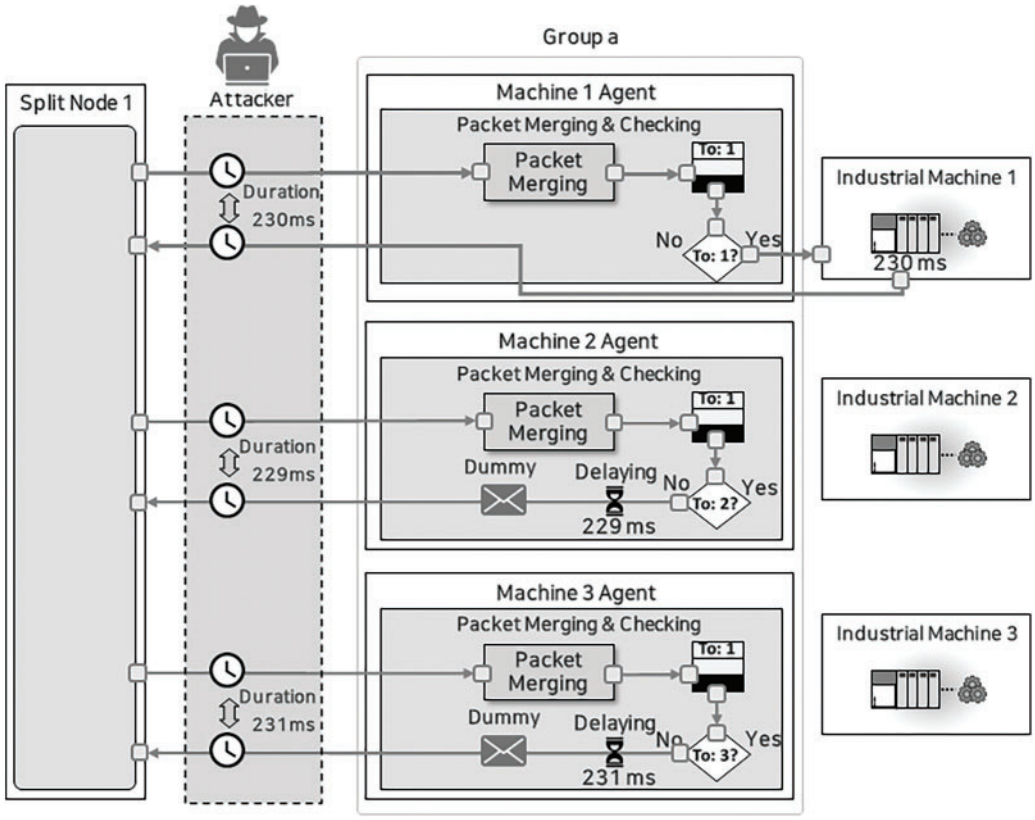

Figure 5: Operation overview for delay-based attacker deception scheme

3.4 Architecture of the Rule-based Network Traffic Obfuscation System

The rule-based NTOS is a method of setting the size of the packets transmitted to the actual destination according to the predetermined rule (Tab. 1) when a node splits an incoming packet into n packets according to the number of group members on the receiving end (Fig. 6). We were inspired by the Moving Target Defense technique [7], which allows access only to the entities knowing the rule shared by exchanging the rules in advance and changing the configuration, and applied it to the NTOS model, drawing on the concept that only those knowing the rule can know the destination and the flow of the original packet.

Table 1: Rule table for traffic obfuscation

Time slot	Packet size
00:00 ~ 01:00	+2
01:00 ~ 02:00	+1
02:00 ~ 03:00	+3
03:00 ~ 04:00	+1
04:00 ~ 05:00	+2
...	...

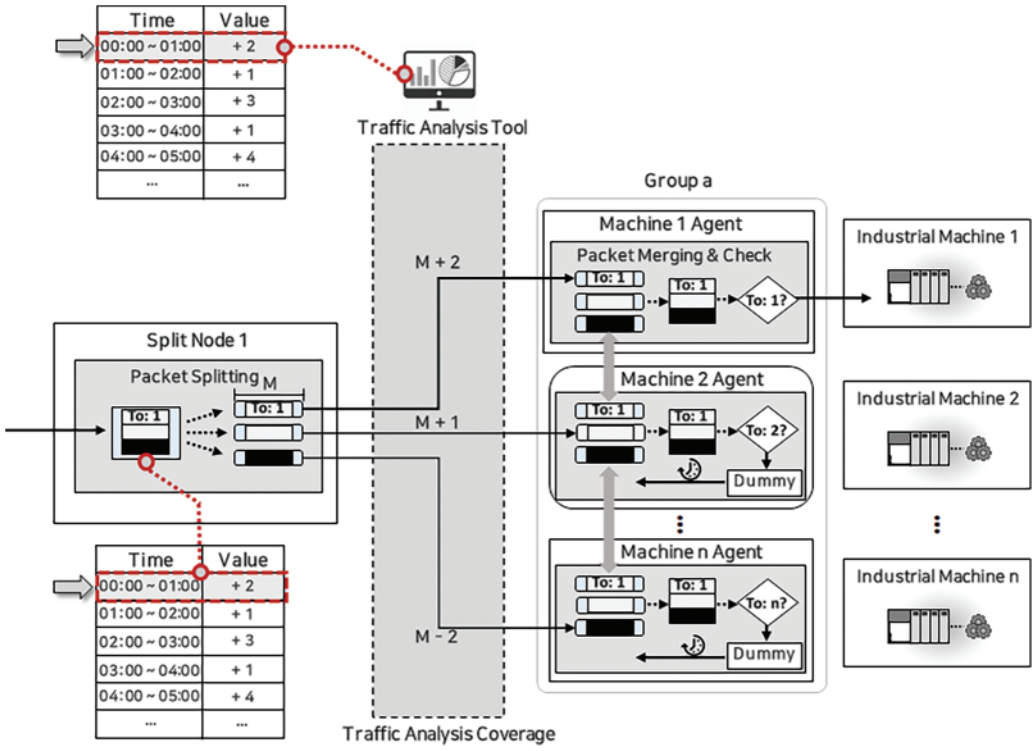

Figure 6: Overview of rule-based traffic obfuscation technique

For the application of the rule-based NTOS proposed in this study, the agents process the packet at the sizes of 256 B, 512 B, and 1024 B. M denotes the size of the split packages of the rule-based NTOS. The split node inserts the padding ranging from 1 to N (= n × 2, the number of group members) into the M sized split packets according to the rule of the rule-based NTOS. For example, if an incoming packet at a split node is 256 B and the number of the destination group members is four, the packet is split into 256 B / 4 = 64 B, where M is 64 B and N is 8 (4 × 2). According to the rule in Tab. 1, split packets are generated with the padding of 64B - 64 + 8B inserted into them. The agent on the receiving end takes the 66 B split packet and the other agents the 64B - 64 + 8B packets minus 66 B. The size of the packets should not overlap. An attacker only knows that the split packets are of different sizes but does not know the underlying rule. The ICS network traffic analyzer, which knows the rule, knows that the destination of the 66 B packet is the recipient agent by size-based traffic analysis. In addition, the risk of the attacker determining the traffic flow by the packet size can be reduced by reducing the time for a configuration change to avert the potential packet size analysis attack.

3.5 Flow Diagram of the Proposed Network Traffic Obfuscation System

Fig. 7 illustrates the detailed flow diagram of the proposed NTOS. It consists of eight states, of which the last three depict the process mirroring the first three as the receiving and sending ends, respectively. Therefore, the process up to State 6 is explained in this section. The flow chart was constructed based on the scenario that Manager 1 sends a packet to Machine 1.

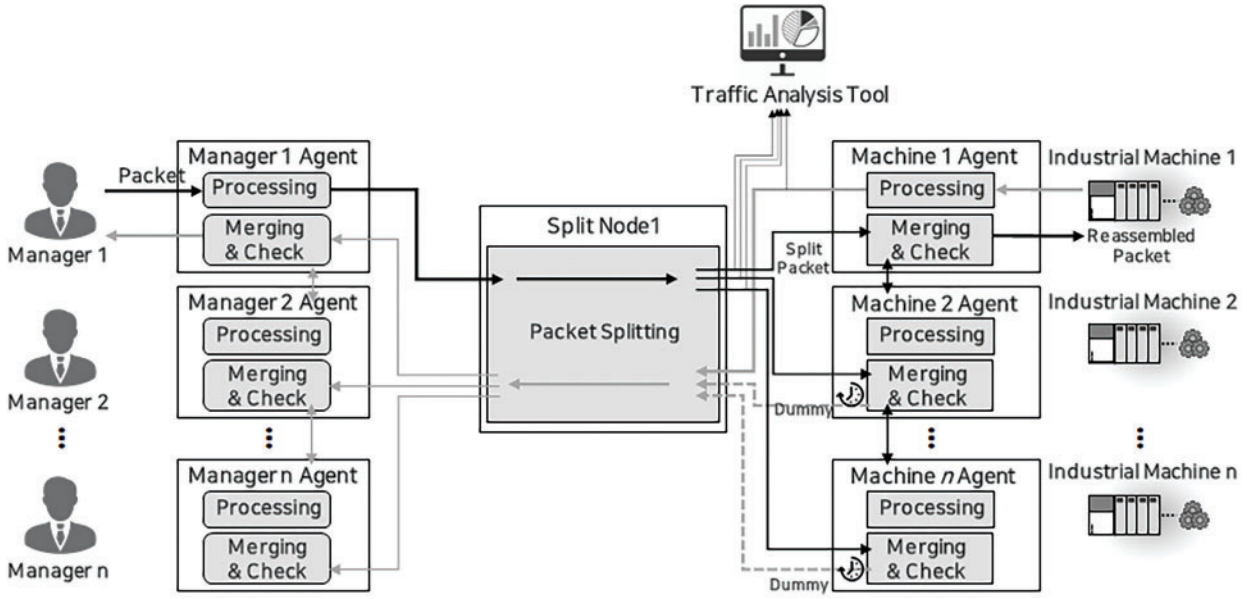

Figure 7: Overall message flow diagram from State 1 to State 8

- **State 1:** Manager 1 generates Message 1–1 to be transmitted to Machine 1 and gives it to the Manager 1 Agent. The Manager 1 Agent, who is performing the transmission for the first time, does not know the symmetric key of the split node and that of the Machine 1 Agent. Therefore, it has to ask the split node to send the private key. The Manager 1 Agent encrypts the IP of Machine 1 and the destination of the message, with the public key of the split node, and generates Message 1–2, and sends it to the split node. The split node decrypts Message 1–2 with its private key to identify the destination IP. Then, it encrypts the symmetric keys of the group to which Machine 1 Agent belongs, and the split node with the symmetric key of the Manager 1 Agent, and transmits Messages 1–3 to the Manager 1 Agent who decrypts Messages 1–3 and obtains the symmetric keys of the split node and the group to which the Machine 1 Agent belongs.
- **State 2:** The Manager 1 Agent encrypts Manager 1's Message 1–1 and the source IP of Manager 1 with the symmetric key of the Machine 1 Agent's group and generates Message 2–1 by encrypting the encrypted message and Machine 1's IP with the symmetric key of the split node. Then, the Manager 1 Agent transmits Message 2–1 to the split node. The split node decrypts Message 2–1 with its symmetric key to identify Machine 1's IP. The split node then again encrypts the decrypted packet along with Machine 1's IP with the symmetric key of the Machine 1 Agent's group and splits the decrypted message into the number of Machine 1 Agent's group members. After inserting the packet identification and sequence numbers to enable the agent group to merge the split packets and restore the message, the split node inserts padding into each of the split packets to a certain size as governed by the rule of the rule-based NTOS, thus generating split messages, and transmitting the split messages to each of the Machine 1 Agent's group members.
- **State 3:** The Machine 1 Agent group members decrypt their respective split messages received with the symmetric key and store the encrypted split messages. Then, they share their respective split messages with all the other group members by broadcasting them over the ICS LAN. The use of LAN enables faster and more effective transmission because it does not burden the WAN

broadband. The ICS components are all adjacent machines and use LAN for efficient internal communication. WAN communication is used only when LAN is not available owing to the addition or removal of machines. The Machine 1 Agent merges split messages into encrypted Message 1–1 using the packet identification and sequence numbers, decrypt it with the group symmetric key, and check the Machine 1 IP and Message 1–1. The Machine 1 agents proceed to State 4 if the destination of the packet coincides with the machine to which they are assigned to or else proceed to State 5.

- **State 4:** The message flow enters State 4 when the destination IP of the decrypted message coincides with the IP of the Machine 1 agent. Machine 1 Agent transmits the decrypted Message 1-1 to machine 1 to which it is assigned. Through the process steps carried out up to now, Machine 1 can receive Message 1–1, the request message from Manager 1. After conducting the request in Message 1–1, Machine 1 generates Message 4–2 as its response or result to be sent to Manager 1 and transmits Message 4–2 to Machine 1 Agent.
- **State 5:** The message flow enters State 5 when the destination IP of the decrypted message does not coincide with the IP of the Machine 1 Agent. The Machine 1 Agent analyzes decrypted Message 1–1 to check the response time for the request exchanged earlier based on the request in the message analyzed. For example, if it takes 100 ms to respond to the request, the Machine 1 Agent sends a dummy message to the split node after holding it back for 100 ms.
- **States 6, 7, 8:** In States 6–8, Machine 1 and the Machine 1 Agent generate messages and transmit them in the same process as carried out by Manager 1 and the Manager 1 Agent in States 1–3. States 7 and 8 are mirror processes of states 2 and 3, respectively.

4 Experiment and Evaluation

The performance of the packet scrambling-based NTOS proposed in this study was experimentally analyzed and assessed. For the performance assessment and experiment, three experiments were conducted based on the requirements: (1) the ability to counter global passive adversaries: comparison of the level of analysis difficulty and transmission speed depending on the traffic volume between the mix node model and the packet scrambling-based NTOS; (2) accuracy of the rule-based NTOS: accuracy test of the rule-based NTOS by analyzing the transmission packet in the network environment where it is applied [27]; and (3) transmission efficiency: measurement of the overhead incurred by the application of the proposed NTOS model.

4.1 Test Environments

To evaluate the level of analytical difficulty and the packet transmission speed of the packet scrambling-based NTOS proposed in this study, we set up a test environment by generating a virtual machine using VMWare. The components of the proposed model were constructed in the form of containers using Docker on VMware. Fig. 8 illustrates the experimental setup and the network setup with each component connected to a LAN or WAN through the Docker network setup. The Manager Agent, Machine agent, and split node operate in a broadband environment assumed as the external network 172.17.0.0/24. The global passive adversary, the attacker model of this experiment, can monitor all the traffic transmitted on the broadband network 172.17.0.0/24. The manager and machine communicate with their agents through a separate internal network. Machine agents of the same group are connected among themselves through the ICS network 172.25.0.0/24. In the experiment, the Manager sends two numbers to the machine as the request, and the machine sends the Manager the number multiplied by two as the response to measure the packet transmission speed; that is, the industrial device responds by multiplying the two numbers.

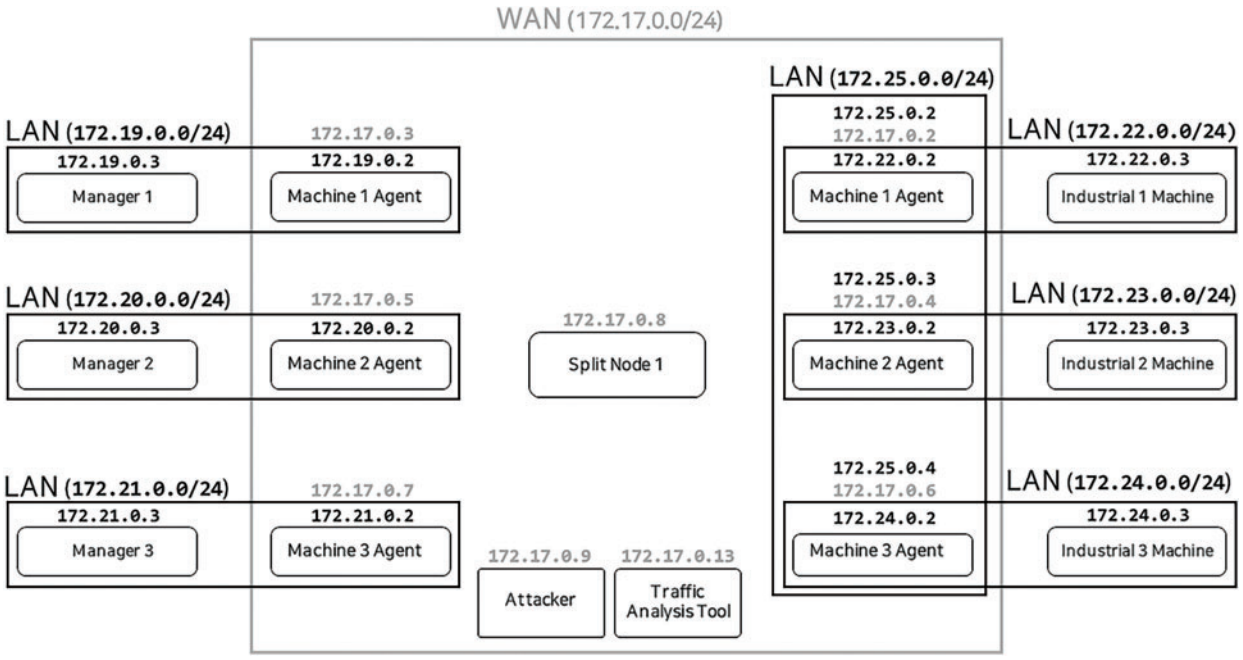

Figure 8: Experimental environment with a network configuration

4.2 Assessment of the Traffic Volume-Dependent Level of Analysis-Difficulty and Speed of the Network Traffic Obfuscation System

We measured the package transmission speed and the traffic analysis probability (level of analysis difficulty) of the conventional timing-based NTOS model and the proposed packet scrambling-based NTOS, varying the packets per second. As a result, it was verified that the packet scrambling-based NTOS did not lose speed or analysis rate with a decrease in the traffic volume compared with the timing-based NTOS.

The analysis probability of the timing-based NTOS is the probability of finding the actual packet between the split packets received and transmitted by the mix node during the delay period. Its response time is from the Manager sending the request to receiving the response from the machine. The traffic analysis probabilities and packet transmission speeds were measured while varying the receiving packets per second (PPS) at the splitting node (172.17.0.8) and mixed node within the range of 250 to 50.

In the experiment shown in Fig. 9, when the analysis probability was assumed to be 1 /5, the timing-based NTOS showed an increase in the delay to maintain the analysis probability at the 1/5 level, which led to an increase in response speed as the PPS decreased. In contrast, the proposed packet scrambling-based NTOS showed no decrease in the response time and analysis probability despite a decrease in the PPS.

In the second experiment, when the timing-based NTOS and the packet scrambling-based NTOS were compared, the former maintained the level of analysis difficulty, despite a decrease in the traffic volume. The experiment was conducted by comparing the images visualizing the packet flows of the timing-based NTOS and the packet scrambling-based NTOS when the Manager 1 Agent (172.17.0.3) sends a packet to the Machine 1 Agent (172.17.0.2)

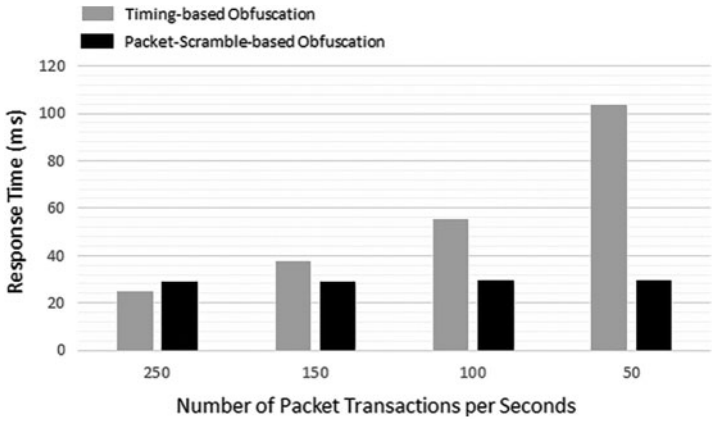

Figure 9: Response time comparison between the timing-based obfuscation and the packet-scramble-based obfuscation schemes

In the timing-based NTOS shown in Fig. 10a, the Manager 1 Agent (172.17.0.3) sends a packet, which passes through three mix nodes (172.17.0.10), (172.17.0.11), and (172.17.0.12) to reach the Machine 1 agent (172.17.0.2) and receives the response. Fig. 10a shows an image visualizing the packet flow of the timing-based NTOS using the network visualization tool EtherApe [28]. When transmitting one packet, even if the delay did not last an infinite time to receive a given number of packets, a clear packet flow was observed because of the method of transmitting the packet after a specified timespan had elapsed.

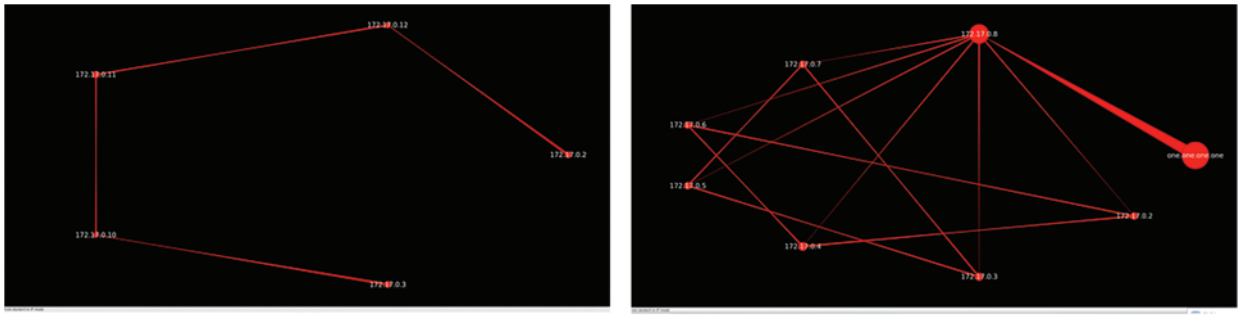

Figure 10: Packet flow for a request between manager-1 agent and industrial device-1 agent (a) in case of timing-based traffic obfuscation (b) in case of scrambling-based traffic obfuscation

In the packet scrambling-based NTOS shown in Fig. 10b, the Manager 1 Agent (172.17.0.3) sends a packet to the Machine 1 Agent (172.17.0.2) through split node 1(172.17.0.8) and receives the response. Fig. 10b shows an image visualizing the packet flow of the packet scrambling-based NTOS using the network visualization tool EtherApe [28]. The packet scrambling method could hide the packet flow by sending it as multiple split packets and transmitting them to multiple agents. The packet scrambling-based NTOS could hide the traffic flow of a single packet as well and could also safely maintain that analysis difficulty level even when the traffic volume decreases. The packet scrambling-based NTOS showed the attacker the analysis difficulty level of (number of manager group members) $\times$ (number of machine group members).

4.3 Experiment and Evaluation of the Rule-Based Network Traffic Obfuscation System

This experiment analyzes the network packets collected by the split node from the perspective of a global passive adversary when split packets are transmitted to the machine agents and the network traffic analyzer that knows the rule of the rule-based NTOS. The results show that the attacker does not know the destination, but the ICS does.

In this experiment, the Manager 1 Agent (172.17.0.3) sends a packet (packet size: 256 B) to the Machine 1 Agent (172.17.0.2) of group a, which has three members. The rule presented in Tab. 1 is applied and the current time is assumed to be 00:00. Tab. 2 outlines the statuses of the split packets transmitted from the split node to the agent group during the communication between the Manager 1 Agent (172.17.0.3) and the Machine 1 Agent (172.17.0.2) at 00:00.

Table 2: Network packet enhanced with the rule-based traffic obfuscation shceme obfuscation

#	Time	Source	Destination	Protocol	Length	Difference
1	**00:00:16.910396**	**172.17.0.8**	**172.17.0.2**	**UDP**	**88**	**2**
2	00:00:16.910442	172.17.0.8	172.17.0.4	UDP	89	3
3	00:00:16.910528	172.17.0.8	172.17.0.6	UDP	87	1
4	**00:00:17.017452**	**172.17.0.8**	**172.17.0.2**	**UDP**	**88**	**2**
5	00:00:17.017509	172.17.0.8	172.17.0.4	UDP	87	3
6	00:00:17.017545	172.17.0.8	172.17.0.6	UDP	89	1

At 00:00, the split node generates packets of 88 B ($256 / 3 = 86 + 2$), to be transmitted to 172.17.0.2 according to the rule in Tab. 1, and the remaining packets of either 87 or 89 by randomly adding 1 or 3 B. The network traffic analyzer knows that at 00:00, there are three group agents and the actual packet is 256, 512, or 1024 B. From this, the values of M i.e., the possible basic values of the three split packets when divided by three, are calculated at 86, 171, and 342 B. The differences between M and the packet sizes are presented in Tab. 2. According to the rule-based NTOS in Tab. 1, the packet with a difference of +2 is the actual packet at the destination at 00:00. Conversely, the attacker who does not know the rule of Tab. 1 cannot know the flow of the actual packet.

4.4 Measurement of Overhead Caused by the Application of the Agent-based NTOS

This experiment was conducted to measure the overhead caused by the application of the agent-based NTOS. The results revealed that the application of the agent-based NTOS reduces the downtime. The experiment was configured for the completion of 100 packets during a one-to-one exchange between the Manager and Machine automatically to be followed by communication, using the agent-based NTOS. Then, we measured the time between completing the 100-packet exchange and when the agent-based NTOS initially exchanged the packet. The result revealed a 2.4 ms overhead between the application of the NTOS and the beginning of the communication, demonstrating that the overhead incurred by the application of the agent-based NTOS is negligible.

5 Conclusions

This study proposed a packet scrambling-based NTOS wherein a packet is split and transmitted to multiple agents as an efficient measure to counter traffic analysis attacks seeking to analyze ICS networks. The packet scrambling-based NTOS passes all communication between the manager and

the industrial machine through a split node that splits packets and transmits the resulting multiple packets to prevent the attacker from determining the actual traffic flow. A conventional network traffic analyzer identifies the actual packet flow based on the rules of the rule-based NTOS. For the design proposed in this study, we identified the limitations of previous NTOS studies by analyzing their advantages and disadvantages then derived the requirements for this NTOS from the characteristics of ICS finally implementing a packet scrambling-based approach founded on these specifications. The results of this study can be summarized as follows: First, conventional NTOS models undergo a decrease in the packet transmission speed and analysis difficulty level when the traffic volume decreases. This leads to performance deterioration in the ICS network, the traffic volume of which is low and irregular. To overcome these limitations, packet splitting is necessary to maintain the analysis difficulty level independently of other packets, thus preventing reductions in the packet transmission speeds and levels of analysis difficulty even when the traffic volume decreases. Second, an ICS operates the network traffic analyzer for security and communication measurement. Therefore, the operation of the network traffic analyzer should be assured during the application of the NTOS. To address this problem, we used the rule-based NTOS wherein the network traffic analyzer that is aware of the rule can identify the actual traffic flow. This function of the rule-based NTOS was verified in an experiment in which the ICS of the network traffic analyzer identified the traffic flow. Third, an ICS contains expensive machines that cannot be modified easily, and system downtime causes significant losses. Therefore, when applying an NTOS to the ICS network, a method to minimize the effect on the ICS components and downtime of the system must be identified. To satisfy these requirements, we designed the NTOS as an agent-based model that, when applied, could minimize the modifications to the ICS components and the downtime. As described above, the packet scrambling-based NTOS proposed in this study can, with minimal modification, prevent any attempts at ICS network analysis when exposed to the outside world on external networks while ensuring the normal operation of the network traffic analyzer. In our follow-up research, we intend to develop an NTOS model that further reduces delay to enhance its performance and wards off active attacks by performing packet splitting and merging at the kernel level. Last but not least, it is a great challenge to apply the NTOS to the actual IIoT-Cloud ICS, because the deploying a new security measure to the conventional infrastructure may cause another side effects or system downtime which can cause enormous losses. Therefore, the future work of this study is to derive the challenging issues and solutions for practically applying the network traffic obfuscation schemes to the actual IIoT-Cloud ICS.

Funding Statement: This work was supported by the Defense Acquisition Program Administration and Agency for Defense Development under the contract UD210029TD.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Sajid, H. Abbas and K. Saleem, "Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.

[2] S. Karnouskos and A. W. Colombo, "Architecting the next generation of service-based SCADA/DCS system of systems," in *Proc. IECON 2011-37th Annual Conf. of the IEEE Industrial Electronics Society*, VIC, Australia, pp. 359–364, 2011.

[3] C. Vijayakumaran, B. Muthusenthil and B. Manickavasagam, "A reliable next generation cyber security architecture for industrial internet of things environment," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 387–395, 2020.

[4] A. Derhab, M. Guerroumi, A. Gumaei, L. Maglaras, M. A. Ferrag *et al.,* "Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security," *Sensors*, vol. 19, no. 14, pp. 3119, 2019.
[5] ICSCER Team, *Recommended practice: Improving industrial control systems cybersecurity with defense-in-depth strategies,* Department of Homeland Security, 2016. [Online]. Available: https://www.ics-cert.us-cert.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf.
[6] L. Ribeiro, J. Barata, M. Onori and J. Hoos, "Industrial agents for the fast deployment of evolvable assembly systems," in *Industrial Agents*. Burlington, MA, USA, Morgan Kaufmann Press, pp. 301–322, 2015. https://doi.org/10.1016/B978-0-12-800341-1.00017-6.
[7] H. Okhravi, T. Hobson, D. Bigelow and W. Streilein, "Finding focus in the blur of moving-target techniques," *IEEE Security & Privacy*, vol. 12, no. 2, pp. 16–26, 2013.
[8] K. E. Hemsley and R. E. Fisher, *History of industrial control system cyber incidents,* Idaho National Lab.(INL), Idaho Falls, ID (United States), 2018. [Online]. Available: https://www.osti.gov/servlets/purl/1505628.
[9] B. Harakrishnan, S. Jason, C. Ryan, R. B.Richard, H. Kathryn *et al.,* "Side-channel analysis for detecting protocol tunneling," *Advances in Internet of Things*, vol. 1, no. 2, pp. 13–26, 2011.
[10] X. Cai, X. C. Zhang, B. Joshi and R. Johnson, "Touching from a distance: Website fingerprinting attacks and defenses," in *Proc. 2012 ACM Conf. on Computer and Communications Security*, NC, USA, pp. 605–616, 2012.
[11] Y. Lee and K. Park, "Traffic analysis obstructing scheme for mission-critical system by encrypting source ip," in *Proc. Domestic Conf. on Next Generation Computing*, Jeju, Korea, pp. 75–78, 2020.
[12] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.
[13] R. Barbosa, R. Sadre and A. Pras, "A first look into SCADA network traffic," in *Proc. 2012 IEEE Network Operations and Management Sym.*, HI, USA, pp. 518–521, 2012.
[14] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes and C. Diaz, "A survey on routing in anonymous communication protocols," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–39, 2018.
[15] S. J. Murdoch and G. Danezis, "Low-cost traffic analysis of Tor," in *Proc. 2005 IEEE Symp. on Security and Privacy (S&P'05)*, CA, USA, pp. 183–195, 2005.
[16] S. Chakravarty, A. Stavrou, Keromytis and D. Angelos, Approximating a Global Passive Adversary Against Tor,Department of Computer Science, Columbia University, 2008. [Online]. Available: https://academiccommons.columbia.edu/doi/10.7916/D82J6QBM.
[17] K. A. Stouffer, J. A. Falco and K. A. Scarfone, "Guide to industrial control systems (ICS) security (Tech. Rep. No. 800-82)," Gaithersburg, MD: National Institute of Standards and Technology (NIST), 2013.
[18] J. Johnson, *Roadmap for photovoltaic cyber security.* Sandia National Laboratories, 2017. [Online]. Available: http://sunspec.org/wp-content/uploads/2017/08/RoadmapforPhotovoltaicCyberSecurity-DraftforReview.pdf.
[19] X. FU, "On traffic analysis attacks and countermeasures," Ph.D. dissertation. Texas A&M University, 2007.
[20] Y. Zhu, X. Fu, B. Graham, R. Bettati and W. Zhao, "On flow correlation attacks and countermeasures in mix networks," in *Proc. International Workshop on Privacy Enhancing Technologies*, Berlin, Heidelberg, Springer, pp. 207–225, 2004.
[21] A. Pfitzmann and M. Waidner, "Networks without user observability," *Computers & Security*, vol. 6, no. 2, pp. 158–166, 1987.
[22] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *Proc. IECON, 2011 - 37th Annual Conf. of the IEEE Industrial Electronics Society*, Melbourne, Australia, pp. 4490–4494, 2011.
[23] L. Pope, Cyber-Terrorism and China. MARINE CORPS COMMAND AND STAFF COLL QUANTICO VA, 2008. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA490725.pdf.
[24] D. WATTS, "Security and vulnerability in electric power systems," in *Proc. IEEE 35th North American Power Sym. Conf.*, Rolla Missouri USA, pp. 559–566, 2003.

[25] D. Kesdogan, J. Egner and R. Büschkes, "Stop-and-go-MIXes providing probabilistic anonymity in an open system," in *Proc. Int. Workshop on Information Hiding*, Berlin, Heidelberg, Springer, pp. 83–98, 1998.
[26] C. Gulcu and G. Tsudik, "Mixing e-mail with babel," in *Proc. Internet Society Sym. on Network and Distributed Systems Security*, CA, USA, pp. 2–16, 1996.
[27] D. Bae and J. Ha, "Performance metric for differential deep learning analysis," *Journal of Internet Services and Information Security (JISIS)*, vol. 11, no. 2, pp. 22–33, 2021.
[28] J. Toledo, *Etherape a live graphical network monitor tool,* 2000. [Online]. Available: http://etherape.sourceforge.net.