Sensors and Materials, Vol. 33, No. 10 (2021) 3415–3424
MYU Tokyo

S & M 2699

Design of Industrial Control System Secure Communication Using Moving Target Defense with Legacy Infrastructure

Jung-Shian Li,[1,2] Chuan-Gang Liu,[3] Chin-Jui Wu,[1,2] Chi-Che Wu,[1]
Che-Wei Huang,[1,2] Chu-Fen Li,[4] and I-Hsien Liu[1,2*]

[1]Department of Electrical Engineering, National Cheng Kung University,
No. 1, University Road, East Dist., Tainan City 701401, Taiwan
[2]Institute of Computer and Communication Engineering, National Cheng Kung University,
No. 1, University Road, East Dist., Tainan City 701401, Taiwan
[3]Department of Applied Informatics and Multimedia, Chia Nan University of Pharmacy & Science,
No. 60, Sec. 1, Erren Rd., Rende Dist., Tainan City 717301, Taiwan
[4]Department of Finance, National Formosa University,
No. 64, Wenhua Rd., Huwei Township, Yunlin County 632301, Taiwan

(Received June 30, 2021; accepted September 21, 2021)

Keywords: industrial control system, moving target defense, secure communication, DHCP, DNS, sensing network

In this paper, we propose a framework that protects the communication for programming logic controllers (PLCs) and sensors in a supervisory control and data acquisition (SCADA) network with an improved moving target defense (MTD) scheme that thwarts attackers in the reconnaissance stage. Our framework changes the Internet Protocol (IP) addresses of each host based on specified time intervals, and the scheme does not need to transmit the IP address to the communication parties for notification. The scheme uses the Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) to improve existing MTD schemes, which may have synchronization problems or a single point of failure. Moreover, adding DNS and DHCP into the MTD scheme significantly lowers the cost of deployment compared with deploying MTD devices before each PLC, making it feasible for an enterprise to implement. Experimental results are presented to demonstrate that our framework can effectively protect a network and that its performance is acceptable.

1. Introduction

Most programming logic controllers (PLCs) and sensors in a traditional supervisory control and data acquisition (SCADA) network are connected in serial communication such as with the RS-232 or RS-485 standard. In this way, the connection is simple and fast. However, such implementation is difficult to realize in larger fields (water utility, power grid, or related PLCs) due to the transmission speed and transmission distance. With industrial control system (ICS) networks gaining in popularity, the communication in ICSs has gradually changed from serial

*Corresponding author: e-mail: ihliu@cans.ee.ncku.edu.tw
https://doi.org/10.18494/SAM.2021.3513

ISSN 0914-4935

communication to Ethernet communication. The advantage of the change is that Ethernet communication supports existing protocols and increases the transmission distance and speed. However, it also brings security risks. Since serial communication is a relatively closed system, the impact is not high when faced with a cyberattack. However, Ethernet communication increases the risk of an attack in devices and sensors.

Information security is becoming increasingly important because of the sophistication of cyberattacks nowadays. Moreover, ICS attacks are also increasing. Hackers attack critical infrastructure to make a profit, which is harmful to both the government and people. For example, in 2015, the Ukrainian power grid was hacked, causing a prolonged power outage. In February 2021, an attacker remotely accessed a Florida water treatment facility to modify chemical levels in drinking water, which could have damaged people's health. In May 2021, an American oil pipeline system suffered a ransomware cyberattack, which led to a fuel shortage. By modifying information in sensing systems, attackers can perform malicious operations on a system. Therefore, ICS security is vital in this era.

Traditionally, firewalls and intrusion detection systems (IDSs) are used to protect ICSs. Some policies are set to avoid attacks. However, vulnerabilities cannot be avoided completely since attackers have unlimited time to break into systems. Moreover, ICSs are often not updated to the latest version to ensure the stability of ICS operation and may suffer from zero-day attacks. Machine learning (ML)-based IDSs have also been studied as a means of protecting IDSs, which also reduce the rate of false detection. However, these methods are all targeted at reducing vulnerabilities and avoiding attacks, which may not be enough for ICS security. Therefore, a moving target scheme has been introduced into ICS security.

Different from the traditional defense, moving target defense (MTD) aims to hide devices to thwart attackers at the reconnaissance stage. MTD changes some communication parameters in the system every time interval. For example, the Internet Protocol version 4 (IPv4) addresses of the communication parties are changed every minute, which makes it difficult for attackers to guess the address of the targeted devices. Even if attackers find the address, it may be changed immediately, and the scheme can be more powerful in Internet Protocol version 6 (IPv6) since the address space is large. To implement the MTD scheme for most SCADA networks, we initially added MTD-enabled modules for each PLC. According to the experimental results, the method is compatible with all devices. However, the deployment cost is relatively high for the related devices. Therefore, in our study, we introduce the Domain Name System (DNS)[(1)] and Dynamic Host Configuration Protocol (DHCP)[(2)] into the MTD scheme. The advantage of introducing DNS is that the deployment cost can be reduced. Thus, the scheme can be applied to sensing networks to serve as a protection layer to prevent attackers from malicious operations. However, the efficiency and security of the communication are somewhat sacrificed in the communication.

The remainder of this paper is organized as follows. In the next section, we describe previous related studies on ICS security including ML-based IDS and MTD schemes. Then the proposed scheme and testing for implementation are presented, followed by a conclusion.

2. Related Work

This section presents studies on ICS security including traditional defenses such as IDS and access authentication. MTD is then introduced, including time-based MTD and mobile IPv6-based MTD. The disadvantages of these approaches are briefly discussed.

Traditional defense in ICSs mainly focuses on access authentication and IDSs.[3] These methods aim to detect anomalous traffic to thwart potential attackers and illegal users. Lu *et al.* proposed a scheme of access authentication for ICSs, which achieved two-way authentication and secured the privacy of communication parties.[4] ML-based IDSs are also becoming increasingly popular because of their low false positive rate and high rate of correct prediction. Elmaaradi *et al.* proposed a new scheme for IDSs based on an ANN. This architecture uses both network-based and host-based IDSs to overcome their respective limitations.[5] Khan *et al.* proposed a method using a hybrid model that takes advantage of the characteristics of ICS communication and uses a three-level detection approach to detect malicious traffic.[6]

To hide the addresses of ICSs in a SCADA network, MTD has been introduced into ICS security. Heydari *et al.* proposed an MTD scheme, Moving Target IPv6 Defense (MT6D), where the communication parties change their addresses without notifying each other.[7] The communication parties calculate the modified addresses with MTD-enabled devices. The MTD-enabled devices utilize dynamic IID obscuration to change their addresses so that attackers cannot discover them. However, a time-based scheme can lead to synchronization problems, resulting in a large communication delay. Therefore, some mobile IPv6-based MTD schemes have been proposed to avoid synchronization problems. In Ref. 8, the mobile IPv6 protocol was used in the MTM6D scheme, which takes advantage of the binding update procedure in the standard mobile IPv6 protocol to notify the communication parties of new addresses. The benefit of the mobile IPv6-based scheme is that devices can communicate with each other while moving. However, the lack of privacy and anonymity is a problem of the scheme since addresses are sent in the network. Therefore, it is possible for the devices to suffer a black-hole attack. MTM6D II[9] was then proposed to improve MTM6D. The scheme improves privacy and anonymity by removing the permanent IP addresses from all packets. However, confusion in communication occurs in a multi-receiver situation. In the scheme, the transmitter must notify the communication parties that their addresses are going to change. However, if one of the receivers does not respond to the transmitter, then a communication delay can occur.

3. Architecture

This section presents the scheme we proposed in a previous paper,[10] a time-based MTD with IPv6 (TMT6D), and discusses the disadvantages of the scheme. The scheme secures a SCADA network by MTD. Figure 1 shows the architecture of the MTD scheme. The source IP address of the packet is 172.16.6.1 and the destination IP address is 172.16.6.2. The packet is then processed by an MTD-enabled device.

The MTD-enabled device of the human–machine interface (HMI) translates the IP address of the packet into IPv6 format by the MTD mechanism program. In the translated IP address of

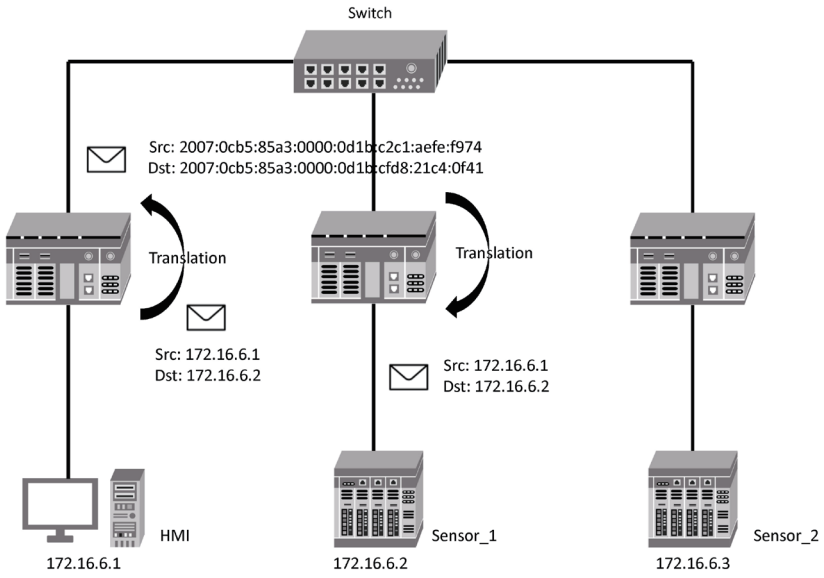

Fig. 1. MTD architecture.

the packet, we take the first eight bits of the host ID as the device identifier. The purpose of the device identifier is to avoid IP address collision. The source and destination IP addresses of the packet are translated to the original address by the MTD-enabled device of sensor_1. The translation not only implements the MTD scheme but also makes the communication transparent.

To improve the synchronization of MT6D, we introduced a modification factor into our MT6D scheme, which is invoked when communication starts. The modification factor specifies the timestamp of the packet and can be divided into two parts: (1) a request part and (2) a reponse part. When the transmitted packet is sent, the modification factor (device time) of the packet is included. The receiver modifies the time using the modification factor and adds the modification factor (corrected time) into the packet as it responds to the transmitter. Below is an example of the modification factor.

In Fig. 2, the communication parties are initially not synchronous. However, with the assistance of the modification factor, the HMI can correct its time on the basis of the device time. In each communication, the communication parties add their timestamps to their packets to correct their time as shown in Fig. 3. However, it is infeasible for an enterprise to implement the above-mentioned scheme because an MTD-enabled device is required for each PLC or HMI, making the deployment cost high. Therefore, in this paper, we propose a DNS- and DHCP-based MTD scheme to lower the deployment cost.

Figure 4 shows the architecture of the proposed scheme. We use DHCP to periodically allocate the IP address of each node device. The communication between two nodes relies on DNS. When DHCP sends a new IP address to a node, it updates the IP table in the DNS server at the same time. The IP table records the IP address corresponding to each node device. If the HMI needs to connect to Sensor_1, it can perform a DNS lookup, and DNS will reply with the address of Sensor_1. Then, the HMI can communicate with Sensor_1.

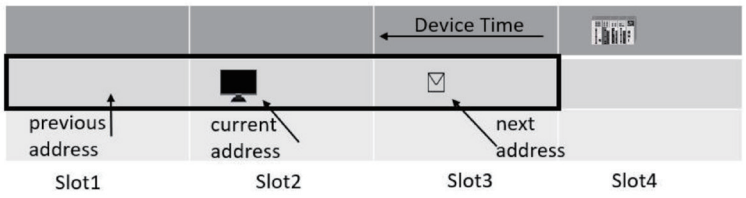

Fig. 2. Modification example (device time).

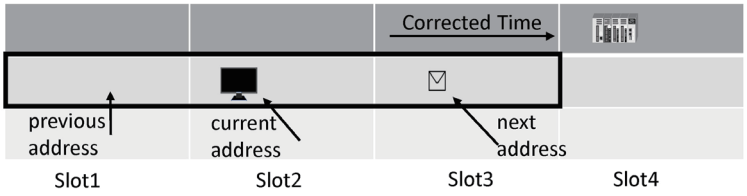

Fig. 3. Modification example (corrected time).

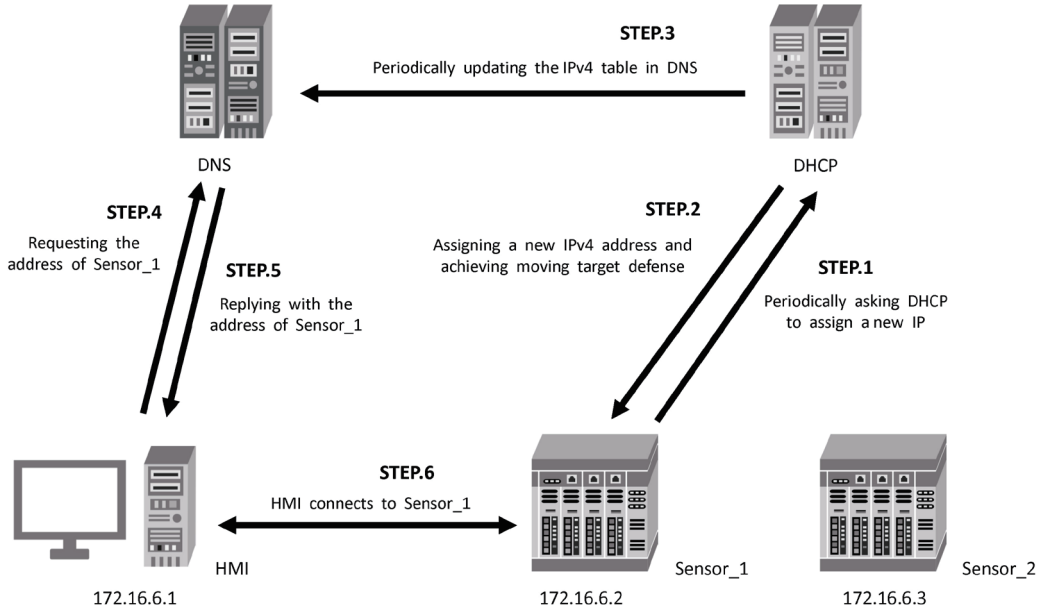

Fig. 4. DNS- and DHCP-based MTD scheme.

The IP address assigned to each node is randomly selected by DHCP from its internal IP pool, and each IP address is unique. Therefore, the situation that two nodes use the same IP address does not arise. In other words, the problem of IP address collision is eliminated. Additionally, each node does not need to use its own timestamp as a parameter to calculate a new IP address, so the time error of the device will not affect communication. The mechanism is performed by one DHCP server, so the problem of time synchronization in MT6D also does not arise in the scheme.

In this mechanism, the cost of implementation is very low, because it is not necessary to connect an external device that is responsible for IP mutation to each node. As a result, we can perform large-scale deployment using this mechanism without a heavy cost.

Now, we explain the operation procedure of this practical MTD architecture. Initially, each device sends out a broadcast packet. The DHCP server replies to each device and assigns an IPv4 address to it. DHCP then updates the IP table in DNS. If the HMI needs to connect to Sensor_1, it sends a request to the DNS server. The DNS server replies with the IPv4 address of Sensor_1. Then, the HMI can connect to Sensor_1. Another feature of this mechanism is the lease term. When DHCP assigns an IP address to a node, it sets an available period for this IP. The IP is returned to the IP pool of DHCP after its expiration, and DHCP assigns a new IP to the node. We can adjust the frequency of IP mutation by changing the lease term of the IP assigned to the node. Figure 5 shows the relationship between the attack and MTD. If the period of IP mutation is shorter than the period of attack, the device will not be attacked. This is realized by setting the lease term to protect each node device effectively.

In summary, this DHCP-based MTD scheme contains the following four features. First, it can be implemented with legacy technologies and can avoid introducing additional problems. Second, the cost of implementing its mechanism is low, making it suitable for large-scale deployment. Third, the problem of IP address collision does not arise. Finally, time synchronization between node devices is not important in this mechanism because we do not use a timestamp while calculating IP addresses. On the contrary, all the IPs are assigned by the DHCP server.

4. Experimental Results

To implement the MTD scheme with MTD-enabled devices, we first construct an environment with Schneider PLCs. In the architecture, users connect to the HMI to request data. The HMI requests the PLCs to acquire data from the sensors through Modbus/TCP.

In the scheme, TMT6D consists of two modules: the IP update module and the IP transfer module. The corresponding pseudocode is provided below. In the program that generates an IPv6 address, we hash the input key and the timestamp of the device and return the processed IPv6 address. In the IP transfer program, we change the IPv4 address into an IPv6 address and transfer the address to the receiver.

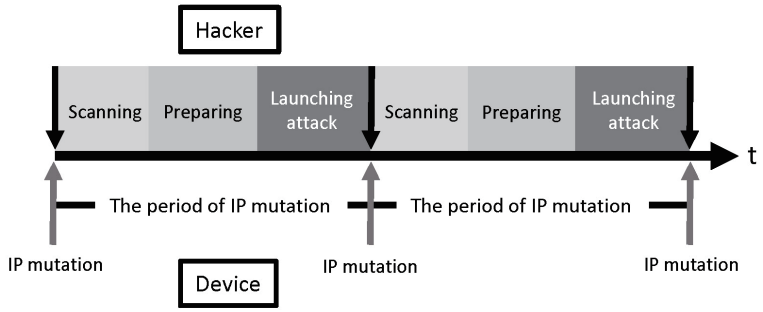

Fig. 5. Schematic diagram of IP-hopping defense.

Algorithms 1-1 and 1-2 give the method of changing the IP addresses of the MTD-enabled devices. The program first requests the operating system to add or delete the IP address through Windows API. Algorithm 1-1 illustrates how the program generates the IP address and ensures its security by the use of a secret key. In Algorithm 1-2, the program generates a task for the system to execute the IP modification in a certain time interval.

Algorithm 1-1
IP address update (generate IPv6 address).

```
1    IP_addr_update(string prefixIP,string secretKey,long time){
2      string Hkey = $"{secretKey}:{time}";
3      byte[] Hbyte = ComputeHash(Hkey);
4      string Hstring = " ";
5      for (int i = 0; i < Hbyte.Length; i++)
6          Hstring += Hbyte[i].ToString("x2");
7      return $"{prefixIP}:{Hstring.Substring(0, 4)}:{Hstring.Substring(4, 4)}:{Hstring.Substring(8, 4)}:
                {Hstring.Substring12, 4)}";
8    }
```

Algorithm 1-2
Dynamic IP address change algorithm.

```
1    Dynamic_IP_change(string prefixIP,string secretKey,long time){
2      while (true){
3        string Skey = "Device_SecretKey";
4        string prefix_IP = "2007:0cb5:85a3:0000";
5        DateTime current_time = DateTime.Now;
6        delIPv6("Eth1", Genertator(prefix_IP, Skey, current_time.AddMinutes(-2).Ticks));
7        AddIPv6("Eth1",Genertator(prefix_IP,Skey,current_time.AddMinutes(-2).Ticks));
8      }
```

Algorithm 2 demonstrates the process of transferring a packet in the MTD-enabled device. First, we use a receiver socket to listen to the message sent by the lower-layer device. Once the receiver socket receives the original IP address, it looks up the corresponding secret key in the table. With the secret key, the MTD-enabled device calculates the destination IP address and builds a transmitter socket to communicate. To authenticate the effectiveness of our MTD scheme, we compare the behavior of our experimental system before and after introducing the scheme. In the first comparison, we observe the return time of packets when the user sends Internet Control Message Protocol (ICMP) packets to PLCs. In the second comparison, we observe the differences in every HyperText Transfer Protocol (HTTP) request of the HMI. The results of the comparison are provided.

Algorithm 2
IP address transfer.

```
1    IP_address_transfer(string srcIP, string dstIP, string message){
2      string dst_PrefixIP = getReceivedPrefixIP(dstIP);
3      string dst_Skey = getReceivedSkey(dstIP);
4      string dst_newIP = Generator(dst_PrefixIP, dst_Skey, DateTime.now);
5      Socket transmitter_socket = new Socket(dst_newIP, 502);
6      transmitter_socket.send(message);
7    }
```

Figure 6 illustrates the difference in the response time of the ICMP test. From Fig. 6, although the return time of the ICMP packets of TMT6D is longer than that of the original packet, the time difference is less than 1 ms. Therefore, we infer that there is little adverse effect on efficiency upon introducing TMT6D. To ensure that the scheme is feasible and does not influence user experiences, we compare the average time of downloading webpages. Figure 7 illustrates the return time of the HTTP request. The effect of introducing TMT6D into the system on users is almost negligible according to the results. However, the deployment cost of the implementation is heavy. Therefore, we proposed the DNS- and DHCP-based scheme to reduce the deployment cost. The results show that the DNS- and DHCP-based scheme has the same efficiency as the original scheme and TMT6D. However, the success of the method depends on the cache time of the computer and browser. If the computer or browser does not query the DNS within the cache time, the communication will fail.

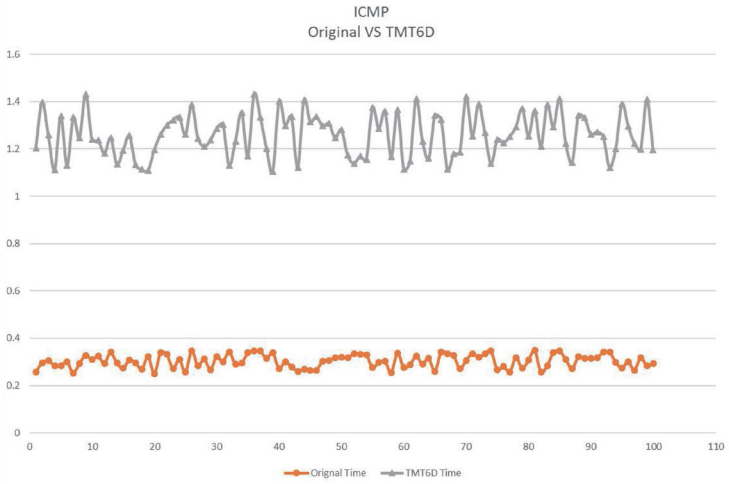

Fig. 6. (Color online) ICMP test.

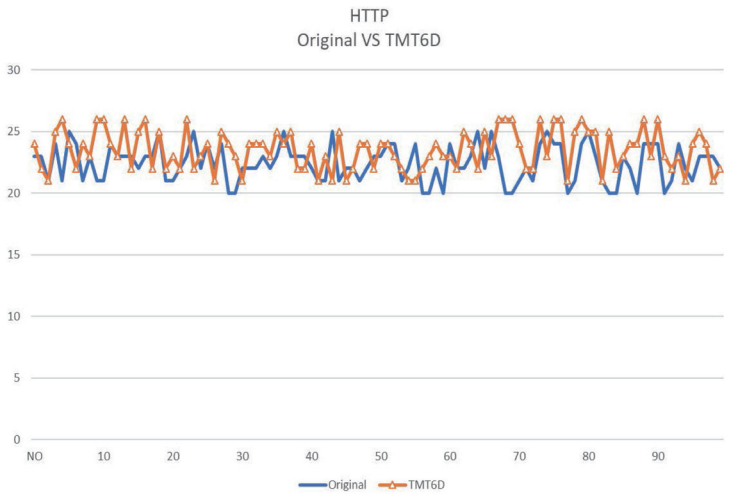

Fig. 7. (Color online) HTTP test.

Table 1
Average response time of Schneider PLCs

	Average response time				
	30 s	10 s	5 s	2 s	1 s
TMT6D-300	68.27 ms	64.13 ms	60.20 ms	62.43 ms	66.41 ms
Normal-300	51.89 ms	51.89 ms	51.89 ms	51.89 ms	51.89 ms
TMT6D-600	75.04 ms	73.26 ms	75.20 ms	75.76 ms	74.34 ms
Normal-600	65.60 ms	65.60 ms	65.60 ms	65.60 ms	65.60 ms

Table 2
Packet loss rate of TMT6D scheme.

	Packet loss rate				
	30 s	10 s	5 s	2 s	1 s
TMT6D-300	0.00%	0.00%	0.00%	0.00%	0.00%
TMT6D-600	0.00%	0.00%	0.00%	0.00%	0.00%

Tables 1 and 2 show the results of 1 min tests. In Table 1, we compare the average response times of adopting the TMT6D scheme and adopting the original PLC communication without any scheme (Normal) in a 1 min test. The first row of the table gives the IP change rate in s, and 300 or 600 in the first column denotes the request rate (requests/min) of the HMI. We can see that the cost of the TMT6D scheme is only around 10 ms. Therefore, hiding the devices from attackers only requires 10 ms. In the MT6D scheme, the packet loss rate gradually increases when the IP rotation interval is less than 10 s. However, in Table 2, we can see that there is no packet loss in the TMT6D scheme even when the IP rotation interval is less than 1 s.

5. Conclusion

Since ICSs have gained increasing attention in recent years, there has been increasing research on the related information security.[11,12] In this study, different from other studies, we propose a method that focuses on solving the problem of vulnerabilities in ICSs. In MT6D, packet loss occurs within a 10 s address rotation interval and address collision may occur. In our scheme, we generate dynamic collision-free addresses. We propose a scheme based on dynamically changing addresses to prevent devices from being found by illegal users and a time modification factor to reduce the problem of time synchronization, and we give pseudocodes for our scheme. According to the results of an experiment, our scheme can achieve similar performance to MT6D, indicating the feasibility of the improved MTD scheme. Moreover, to reduce the deployment cost, we propose a DHCP- and DNS-based architecture for implementation in the real world. In this paper, we give the complete operation procedure of this practical architecture. However, the scalability of the MTD scheme still needs to be tested while implementing such new architecture. In the future, we will carry out such testing so that our scheme has high performance in the real world.

Acknowledgments

This work was supported by the Ministry of Science and Technology (MOST) in Taiwan under contract numbers MOST 109-2218-E-006-014-, MOST 110-2218-E-006-013-MBK, and MOST 109-2221-E-041-001-.

References

1 Domain Names - Implementation and Specification: https://datatracker.ietf.org/doc/html/rfc1035 (accessed August 2021).
2 Dynamic Host Configuration Protocol for IPv6 (DHCPv6): https://datatracker.ietf.org/doc/html/rfc3315 (accessed August 2021).
3 Y. Yang, K. McLaughlin, T. Littler, S. Sezer, and H. F. Wan: Proc. 2nd IET Renewable Power Generation Conf. (IET, 2013) 9–11. https://doi.org/10.1049/cp.2013.1729
4 Y. Lu, X. Chen, and C. Chen: Proc. 2016 Int. Conf. Security of Smart Cities, Industrial Control System and Communications (IEEE, 2016).
5 A. Elmaaradi, A. Lyhyaoui, and I. Chairi: 2019 3rd Int. Conf. Intelligent Computing in Data Sciences (IEEE, 2019). 1–5. https://doi.org/10.1109/ICDS47004.2019.8942383
6 I. A. Khan, D. Pi, Z. Ul. Khan, and Y. Hussain: IEEE Access **7** (2019) 89507. https://doi.org/10.1109/ACCESS.2019.2925838
7 M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront: Proc. 2011 Military Communications Conf. (IEEE, 2011) 1321–1326. https://doi.org/10.1109/MILCOM.2011.6127486
8 V. Heydari, S. Yoo, and S. Kim: Proc. 2016 IEEE Conf. and Exhibition on Global Telecommunications (IEEE, 2016) 1–6. https://doi.org/10.1109/GLOCOM.2016.7842255
9 V. Heydari: IEEE Access **6** (2018) 33329. https://doi.org/10.1109/ACCESS.2018.2844542
10 C. Liu, C. Wu, I. Liu, C. Wu, and J. Li: Proc. 2020 Int. Conf. Intelligent Computing and its Emerging Applications (ACM, 2020) 1–6.
11 S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati: IEEE Commun. Surv. Tutorials **22** (2020) 1909. https://doi.org/10.1109/COMST.2020.2982955
12 X. Zhou, Y. Lu, Y. Wang, and X. Yan: Proc. 2018 IEEE 3rd Int. Conf. Image, Vision and Computing (IEEE, 2018) 821–827. https://doi.org/10.1109/ICIVC.2018.8492800