IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 24, 2018, accepted May 30, 2018, date of publication June 20, 2018, date of current version July 6, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2844542

Moving Target Defense for Securing SCADA Communications

VAHID HEYDARI
Computer Science Department, Rowan University, Glassboro, NJ 08028, USA
(e-mail: heydari@rowan.edu)

ABSTRACT In this paper, we introduce a framework for building a secure and private peer to peer communication used in supervisory control and data acquisition networks with a novel Mobile IPv6-based moving target defense strategy. Our approach aids in combating remote cyber-attacks against peer hosts by thwarting any potential attacks at their reconnaissance stage. The IP address of each host is randomly changed at a certain interval creating a moving target to make it difficult for an attacker to find the host. At the same time, the peer host is updated through the use of the binding update procedure (standard Mobile IPv6 protocol). Compared with existing results that can incur significant packet-loss during address rotations, the proposed solution is loss-less. Improving privacy and anonymity for communicating hosts by removing permanent IP addresses from all packets is also one of the major contributions of this paper. Another contribution is preventing black hole attacks and bandwidth depletion DDoS attacks through the use of extra paths between the peer hosts. Recovering the communication after rebooting a host is also a new contribution of this paper. Lab-based simulation results are presented to demonstrate the performance of the method in action, including its overheads. The testbed experiments show zero packet-loss rate during handoff delay.

INDEX TERMS SCADA, mobile IPv6, moving target defense, dynamic IP.

I. INTRODUCTION

Critical infrastructure, including electricity distribution, water treatment, petroleum refining, etc., is the backbone of our nation's economy, security, and health. Supervisory Control and Data Acquisition (SCADA) systems perform critical functions in controlling industrial systems. A SCADA system includes two main components, a Human Machine Interface (HMI) and a Programmable Logic Controller (PLC). HMI is a user interface for signaling and controlling the state of the system. PLC is directly connected to the physical infrastructure through sensors and actuators. The SCADA system uses a client/server communication model in which the HMI is the client that continually sends write and read commands to the PLC that is the server. In this way, the HMI can send control parameters or read sensor measurements and the state of the PLC program. Cyber-attacks on such infrastructure can cause loss of life, threaten public safety/national security, or impact environmental disasters.

Cyber-attacks can be used to execute injection, replay, alteration, exploits, and Denial-of-Service (DoS) attacks [1]. A cyber-attack may affect a large blackout, disable the safety monitoring system of nuclear power plants [2], or damage the system (e.g., STUXNET [3] and HAVEX [4]). The SCADA Strangelove project [5] identified 150 zero-day vulnerabilities in SCADA systems. Given these examples, it can be seen that improving the security of SCADA systems is very crucial.

In this project, we assume that SCADA networks are penetrable. The next step after penetrating to the network is finding and fingerprinting the PLC(s). After finding targets, cyber-attacks can occur on availability and/or integrity of the system [6]. These attacks can prevent remote monitoring and controlling by a legitimate operator. They can also fabricate, alter, and/or replay network packets between the PLC and the HMI [7]. To prevent these cyber-attacks, we propose a framework for building a secure and private peer to peer communication with a novel Mobile IPv6 based Moving Target Defense (MTM6D) strategy [8]–[11] to prevent remote attacks by IP address hopping. Our approach aids peer hosts to combat remote cyber-attacks by thwarting any potential attacks at their reconnaissance stage. In MTM6D, we utilized Mobile IPv6 [12], where there is a permanent IP address—Home Address (HoA)—which is used to avoid disrupting TCP sessions and a temporary IP address—Care-of Address (CoA)—which is used to connect to other nodes. MTM6D dynamically changes the CoA of a host for moving the target. Note that we treat the hosts as if they were mobile nodes of Mobile IPv6. MTM6D needs a small modification in the standard Mobile IPv6 protocol. Providing dynamic IP addresses only on one node among two connected nodes is the major shortcoming of MTM6D. More clearly, MTM6D cannot protect both peers against remote attacks in the problem subject to investigation. Another shortcoming of MTM6D is incurring significant packet loss (on high latency communication links) during address rotations. Lack of privacy and anonymity for communicating hosts is also another shortcoming of MTM6D. Note that the permanent IP address should be stored in the home address option/the routing header type 2 (IPv6 headers) of each data packet that shows the HoA of the source/destination. Therefore, Man-In-The-Middle attacks that need to target specific IP address(es) and other types of attacks against node's privacy can occur. In this project, a **new version of MTM6D (MTM6D II)** is proposed to resolve the above shortcomings. Furthermore, we propose a way for preventing black hole attacks, as a part of DoS attacks (in which a compromised router on the path between two hosts discards packets instead of forwarding them) and bandwidth depletion Distributed Denial-of-Service (DDoS) attacks (that only need the subnet ID instead of the exact IPv6 address of a target). A method is also presented to recover the communication after rebooting a host.

The proposed method (MTM6D II) is designed to meet the following requirements:

- A static IP address is needed to be transparent to the upper layers. However, the static IP address should not be accessible through the Internet. In this way, a dynamic IP address should be used for connecting to the peer node.
- Changing the dynamic IP address should not cause any delay or packet loss in the network.
- Rotating IP addresses should be done independently on each node. Therefore, a mechanism is needed to update the peer node with the new IP address.
- The new method should also support dynamic address rotation intervals such as a shorter rotation interval during suspicious activity and a longer one to decrease the overhead.
- Adding new requirements or any change in the network equipment should be avoided.
- Mobility between subnets that changes the prefix of IP addresses should be supported.
- A combination of standard protocols should be used instead of creating a new protocol given the point that the new protocol can add new vulnerabilities and may have security or scalability problems.

The last requirement listed above has an essential role in security and scalability. We will show that the proposed method uses Internet Protocol Security (IPsec) with Internet Key Exchange version 2 (IKEv2) [13] instead of defining a new protocol. For example, researchers used covert channels to provide some level of authentication [14], used TCP Option field to carry authentication information [15], or embedded encryption [16], [17] for Modbus/TCP. However, IPsec with IKEv2 is already providing encryption, authentication, key distribution/rekeying, and replay attacks protection. As a result, the proposed method does not depend on a specific algorithm or key size for encryption, authentication, and key distribution. This portability feature helps us to implement this technique for different applications like small low-power Internet of Things (IoT) devices. For example, the choice of cryptographic algorithms is left to negotiation steps of IKEv2 to select an algorithm that both parties support.

The remainder of this paper is organized as follows. In the next section, an overview of the related work is provided followed by the threat model. Then, some details of Mobile IPv6 are explained. Then the proposed solution and results of testing with a prototype implementation are presented followed by our summary/conclusions.

II. RELATED WORK

This section includes a brief review of previous MTD-based methods that protect servers against remote attacks. Also, some of the limitations of these methods are discussed.

Some cloud-based defense methods were presented in [18]–[20], and [21] for Internet services against DDoS attacks. These solutions leverage the on-demand availability of resources in a cloud environment to hide the server's location behind a large pool of proxies. Incoming connection requests from authorized users are redirected by a central server to these proxies to serve the users subsequent requests. When under attack, the central server instantiates new proxies and moves the users associated with attacked proxies to these new proxies. To prevent next attacks, the central server also shuffles the client-to-proxy assignment to isolate insiders who shared the location of the proxies with attackers.

Other cloud-based defense methods, those are based on Virtual Machine Live Migrations (VM-LM), focused on the integrity of software before migration [22] or considered the availability and duration of migration in practice [23]. TALENT [24] is designed for critical infrastructure applications by migrating to a different platform at random time intervals when a new vulnerability or attack is discovered. A security model to assess and compare the effectiveness of these cloud-based MTD methods is presented in [25]. These methods are treatments in nature, instead of prevention. Detecting flooding attacks could be possible by traffic analysis techniques like [26] and [27]. However, it is difficult to detect penetration attacks like remote exploits that take advantage of target vulnerabilities. Therefore, we need to consider both prevention and treatment to provide an effective security scheme.

An MTD technique called OpenFlow Random Host Mutation (OF-RHM) is introduced in [28]. In this technique, an address range, selected from the unused network address space, is assigned to each host. A virtual IP address is chosen from this range at each mutation interval. A Software-Defined Networking (SDN) approach is used for range allocation and mutation coordination. A centralized controller (NOX) establishes flows in OpenFlow switches to forward requests and perform the address translation actions. The virtual IP addresses will be used for routing and are automatically translated into the real IP addresses and vice versa at the network edges (subnet) close to the source/destination. As the advantages of this method, it is transparent to the end hosts and does not use any encapsulation method. On the other hand, the limitations of this method are requiring central management and new equipment and not supporting mobility.

One of the prevention methods is MT6D [29]. MT6D is a form of a dynamic network layer MTD that rapidly changes IPv6 addresses of both the sender and receiver mid-session without dropping or renegotiating sessions. The design takes advantage of IPv6 networks allowing nodes to bind new IPv6 addresses seamlessly. MT6D creates dynamic Interface IDentifier (IID) obscuration to develop dynamic IP addresses. These IIDs are comprised of three parts: (1) a value specific to an individual host (seed IID), (2) a secret (symmetric) key shared between both parties, and (3) a variable that is agreed upon by both sides (e.g., time). Out-of-band is suggested for sharing the seed IID and the key. In this method, peers use the same algorithm with a pre-shared symmetric key that generates a random IPv6 address per each time interval based on the Media Access Control (MAC) address as input. Using the peer's MAC address as input is the way to find the peer's IP address during the current time interval.

MT6D encapsulates original data packets to Unreliable Datagram Protocol (UDP) packets to hide the original IP addresses and uses virtual IP addresses.

The limitations of MT6D for our purpose are as follow:

- Mobility is not supported by this method. For example, if one host moves to a new subnet (or switches between two Internet connections), the prefix of its IPv6 address is changed, and the peer host cannot find it. Note that we need mobility support to prevent black hole and bandwidth depletion DDoS attacks by switching between multiple Internet connections.
- Packet loss due to address collision exists. As the IP addresses are dynamically changed, address collision may occur. Although, because of huge availability of IP addresses in IPv6, the likelihood of an address collision is minimal, the connection will be lost during the rotation interval that an address collision occurs.
- Key management is lacking. Rekeying is needed to improve the security, but MT6D lacks support for key management protocols.
- Relatively tight time synchronization is needed.
- Dynamic address rotation interval is not supported. We may need to change the address rotation interval depending on our network situations. For example, a shorter rotation interval when suspicious activities are detected is preferred, and a longer one is suitable at other times to decrease computational (or network) overhead.

Please note that MTM6D II does not have any of the limitations mentioned above. For example, MTM6D II is based on Mobile IPv6 (to support mobility), uses IPsec/IKE_v2 (to support rekeying), does not need any time synchronization methods, supports dynamic address rotation interval, and has zero packet loss rate.

III. THREAT MODEL

We now discuss the threat model. The focus of this research is on preventing remote cyber-attacks against IP based SCADA protocols. Modbus/TCP, DNP3, Profinet, and EtherNet/IP are the main SCADA protocols that operate over IP. Modbus/TCP [30], operates over TCP/IP, is a member of Modbus protocol family. It was originally developed in 1979 and then become an open standard. It has a simple client/server communication messaging service for requests and responses. The Distributed Network Protocol (DNP3) [31] is a set of communications protocols transported across various physical media, including TCP/IP networks. Its primary use is in utilities such as electric and water companies. Profinet [32] is a standard for data communication over Industrial Ethernet with strength in delivering data under tight time constraints. It leverages TCP/IP for collecting data and controlling equipment in industrial systems. The Ethernet Industrial Protocol (EtherNet/IP) [33] also operates over IP. It uses its object-oriented design and adapts the Common Industrial Protocol. EtherNet/IP makes use of both TCP and UDP for explicit and implicit messaging. As the proposed method in this paper is implemented in the network layer (IP layer), it is compatible with all IP based SCADA protocols mentioned above.

Remote cyber-attacks include special actions which allow attackers to compromise remote systems. Address-based DDoS attacks and remote exploits are two main categories of remote attacks that need to know the IP address of their intended target(s). Remote exploits take advantage of a bug or vulnerability to view or steal data or gain unauthorized access to a vulnerable target.

For this research the cyber-attacks from four categories are considered; enumeration, confidentiality, integrity, and availability.

A. ENUMERATION

Step one of a cyber-attack is finding a target. Network mapping tools, such as NMAP, can be used to search IP address ranges for connected devices. For example, finding open ports associated with industrial control system communication protocols; Modbus/TCP, DNP3, Profinet, EtherNet/IP, etc. can help attackers to find PLCs. Once a PLC is located, additional enumeration techniques are available to fingerprint the device. This type of information can be used to identify specific vulnerabilities to exploit against systems.

B. CONFIDENTIALITY

Attackers may eavesdrop on network communications between SCADA components to learn details of system operation, and reverse engineer experiment construction.

C. INTEGRITY

Most SCADA networks do not employ tunneling technologies such as IPSec or SSL to provide network packet integrity at higher network layers either. As such, network packets carrying sensor measurements and supervisory commands can be altered, replayed, or crafted and injected into the network.

D. AVAILABILITY

Network availability is considered critical for SCADA systems because the network is needed both for monitoring and controlling the remote physical process. As such denial of service vulnerabilities are considered significant threats to SCADA systems.

IV. BACKGROUND

In this section Mobile IPv6, stateless address autoconfiguration, route optimization, binding management, and multiple CoAs registration are introduced. These concepts are essential for understanding the rest of the study.

A. MOBILE IPv6

Mobile IPv6 is utilized as the base of the proposed method to take advantage of several of its features. One of the most important features is its ability in handling the changing IP address of a Mobile Node (MN) as it moves to other subnets. Though we do not have real mobility, we treat both parties as MNs. In Mobile IPv6, an MN has two different IP addresses. One of them is a permanent IP address, HoA, assigned by the Home Agent (HA) and another one is CoA, which is used by others, called correspondent nodes (CN), to reach the MN. HA is a router on the MN's home link that functions similar to a proxy for the MN and keeps track of the CoA and performs the necessary forwarding. When the MN moves between subnets and changes its CoA(s), it should update the HA using Binding Update (BU) messages that contain new CoA(s). In response, Binding Acknowledgement (BA) messages can be used to make sure that the HA is updated. In the default implementation of Mobile IPv6, CNs contact the MN via its HoA, which is processed by the HA and tunneled to the MN.

B. STATELESS ADDRESS AUTOCONFIGURATION

IPv6 hosts can use Neighbor Discovery protocol via the Internet Control Message Protocol version 6 (ICMPv6) Router Discovery messages for autoconfiguration when they are connected to an IPv6 network [34]. Hence, each host can automatically generate global IPv6 addresses without needing any manual configuration or help of a server. The neighbor discovery protocol provides powerful mechanisms that allow hosts to obtain all the necessary information about their link. Autoconfiguration is started by generating a link-local address for the network interface (tentative address). Then a Neighbor Solicitation message with the tentative address as the target is used to check against current occupancy of the tentative address. If this address is occupied, the host will receive a Neighbor Advertisement message. Therefore, this tentative address cannot be employed and another address should be generated, and the same process should be repeated.

C. ROUTE OPTIMIZATION

Route optimization is used to forward packets directly between an MN and a CN (or another MN). In order for this strategy to work, the CN should hold the MN's current CoA. Therefore, the MN should update the CN with the latest CoA. Before this direct communication, return routability procedure [12] should be used to verify the right of the MN to use a specific HoA and to verify the validity of the claimed CoA. This procedure involves four messages. Following this process, two additional messages (BU and BA) will be sent.

After running the route optimization mechanism, packets will be forwarded directly between two MNs (in the proposed method both parties are MNs). More specifically, the source and destination IP addresses in each packet's header are CoAs of MNs. However, to be transparent to the upper layers, HoAs (permanent IP addresses) should be in the packet's header and swapped with CoAs in the source and destination. For this purpose, Routing Header Type 2 and Destination Options Header are defined in Mobile IPv6 [35].

The routing header type 2 is used by an HA or a CN to carry the MN's HoA when packets are sent to the MN's CoA. For example, after the route optimization mechanism, a CN (or another MN) knows the MN's CoA so that the CN can send a packet directly to the MN's CoA, but the MN needs to see its HoA in the destination IP address. Therefore, the CN stores the MN's HoA in the routing header type 2 and the MN's CoA in the destination IP address and forwards the packet. When the MN receives this packet, it automatically swaps the destination IP address of the packet with the address stored in the routing header type 2.

The destination options header is used to carry optional information that needs to be processed only by the destination node. Home address option is an essential part of this option. It is used in packets sent by the MN while away from home, to inform the CN (or another MN) of the MN's HoA. In turn, the source address of the packet is the CoA of the MN, and the HoA of the MN is stored in the home address option. When the destination receives this packet, the MN's CoA and HoA will be swapped if the pair of the CoA and the HoA of the sender is found as a Binding Cache entry that is explained in the following.

D. BINDING MANAGEMENT

In this subsection, Binding Update List and Binding Cache are explained as two data structures needed for direct communication between an MN and a CN (or another MN) based on the route optimization mechanism.

1) BINDING UPDATE LIST [12]

Each MN has a Binding Update List (BUL). The BUL records information for each BU message sent by this MN that includes all bindings sent by the MN to its HA or CNs (or other MNs). When a new BU message is sent to the same destination, the entry of the BUL that stored information about the previous BU message will be updated with this new BU message. When a new packet is ready to be sent, information about the BUL is the key to decide whether this packet should be sent to the destination directly or via the HA.

The most important fields of each BUL entry are:

- The IP address of a CN (or the HoA of another MN) to which a BU message was sent.
- The HoA for which that BU message was sent.
- The CoA sent in that BU.

2) BINDING CACHE [12]

Binding Cache, which includes bindings for other nodes, is used for the route optimization mechanism. Main fields of Binding Cache are:

- The HoA of the MN for which this is the Binding Cache entry. This field is the key for searching the Binding Cache to find whether an entry exists for this destination or not.
- The CoA for the MN indicated by the HoA field in this Binding Cache entry.

When the MN registers a new CoA, it will subsequently send BU messages to all of its CNs listed in the BUL. Note that as a default in the standard of Mobile IPv6, the MN does not check the Binding Cache when it wants to send BU messages. That means, BU packets have the destination option header but do not have routing header type 2.

E. MULTIPLE COAS REGISTRATION

One of the keys to having zero packet loss rate when an MN changes its CoA for moving the target, is the ability to have multiple CoAs at the same time. According to the multiple CoAs registration rules of Mobile IPv6, an MN can utilize multiple CoAs (over the same HoA) with its HA and/or CNs. The MN automatically sends BUs to its HA and/or CNs per each new IPv6 global address that has been registered as its CoA.

V. PROPOSED SOLUTION

As discussed before, the focus of this paper is on preventing remote attacks against two hosts connected through the Internet. Towards this goal, we leverage a network layer moving target defense to avoid each host being targeted for exploitation. Note that each host can be a computer, a low-power IoT device, a network gateway, a PLC or an HMI of SCADA systems.

To better understand the need for the proposed method, it needs to be answered why we need an MTD method when we have other defensive measures like IPsec. The MTD method is necessary because IPsec with an Internet key exchange method, like IKEv2, is a computer program that could be threatened by zero-day vulnerabilities. For example, a UDP port needs to be open to start IPsec/IKE. This open port can be targeted by DoS attacks or buffer overflow vulnerabilities [36]. Therefore, one efficient way is preventing a system from being targeted for attacks, i.e., preventing attacks at the reconnaissance step instead of letting attackers to find the system and its open ports and start testing different ways to penetrate. More clearly, the proposed idea is moving around (in the vast address space of IPv6) as fast as possible instead of staying in the same place to be targeted by attackers. Note that we use IPsec/IKEv2 in the proposed method as a Defense in Depth and to remove permanent IP addresses to improve privacy and anonymity. However, attackers need to find the dynamic IP addresses of our systems in the first step, a task which is not easy to accomplish. For example, with less than half a second round-trip time between two hosts, we can dynamically change their IP addresses in an unpredictable way every two seconds. This way, attackers need to find the current IP addresses between about 18 quintillion choices (if we use 64 bits of IPv6 address size for the interface identifier) in less than two seconds. This is nearly impossible with currently existing computing and network resources. Therefore, even an unpatched implementation of IPsec/IKE with well-known vulnerabilities cannot be easily targeted in the proposed method.

Mobile IPv6 [12] is employed as the base of the proposed method. The hosts (hereafter referred to as MN1 and MN2) act like MNs of Mobile IPv6. HoAs of the MNs are used as the permanent IP addresses to be transparent to the upper layers. CoAs of the MNs are used as the dynamic IP addresses of the hosts. Random IP address rotator is implemented to change the CoAs for moving targets dynamically. Other reasons that Mobile IPv6 is selected are:

- Mobile IPv6 enables each host to cache the binding of a permanent peer's IP address with its dynamic IP address and then send all packets destined for the peer directly to it using this dynamic IP address.
- Binding update mechanism is used to inform each host of the current peer's dynamic IP address.
- Hosts use the new peer's dynamic IP address only after receiving the BU message from the peer. So this new IP address has already been successfully registered by the peer. Therefore, there is no chance for packet loss due to address collision.

Note that accessibility of HoAs (permanent IP addresses) through the Internet leaves the hosts vulnerable to be targeted. This accessibility is only possible via the HAs. Therefore, the HAs should be removed in the proposed method. However, the HAs are needed for the return routability procedure to test the HoAs. To solve this issue, we should use another method that does not require the return routability procedure. For this purpose, as another contribution of this work, we utilize RFC 4449 [37] along with IPsec and IKEv2 in order to create a Secure Route Optimization (SRO) method without the HA participation. SRO is not only useful for our

MTD but also can be used for other applications of Mobile IPv6. More specifically, if the MN is trustable, SRO can be employed to decrease signaling overhead and remove the HA participation. This new method is explained in details next.

In RFC 4449, a static shared key method is presented to omit all messages related to the return routability procedure. We leverage this approach, which results in significant improvements. An MN can update the peer with a new CoA directly because the HA is not involved in the route optimization mechanism. Another improvement is decreasing signaling overhead because only BU and BA packets are needed. Along with these advantages, the static shared key method also has some limitations:

- The peer needs to trust the actions of the MN and needs to assume that the MN will not launch flooding attacks against a third party as described in [38].
- Static shared symmetric keys between the peer hosts are needed. Therefore, this method cannot resist replay attacks.

To solve the first issue, we assume that both parties are trustable. This is not a restrictive assumption, as the peers in the problem subject to investigation are actually trustable. To address the second issue, we combine RFC 4449 with IPsec and IKEv2 between both parties because IKEv2 can provide automatic key distribution/rekeying and protection against replay attacks.

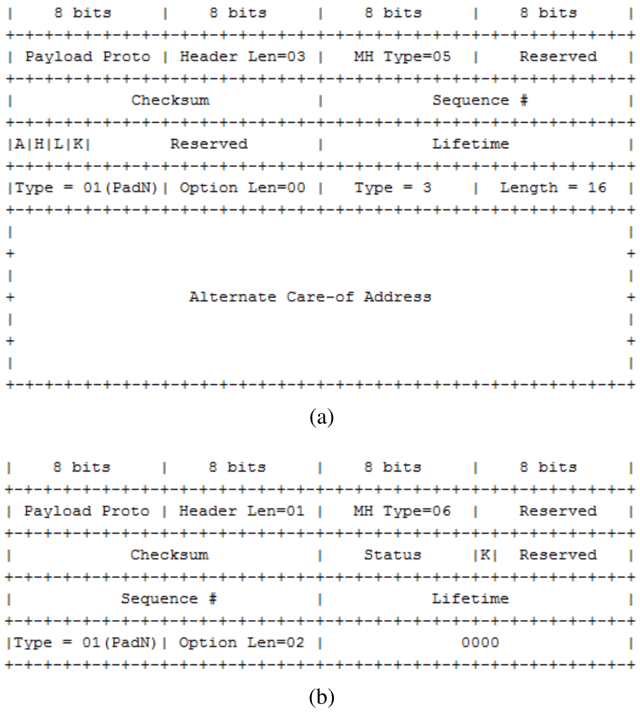
In SRO, BU/BA messages are protected by IPsec, so the binding authorization data (and nonce indices options) are not needed in the mobility header (the extension header of IPv6) [12]. However, the receiver needs a way to verify the claimed identity (CoA in the source of IPv6 packet) of the sender. We have two solutions for this authentication requirement:

- Using Authentication Header (AH) besides Encapsulating Security Payload (ESP).
- Using Alternate Care-of Address option for BU messages encrypted by ESP.

As the default, we propose to use IPsec ESP in transport mode for encrypting both signaling and data packets. Therefore, between the two options, the second one has a better performance. In this way, a copy of the CoA, which is used in the source of IPv6 packet, is automatically encrypted by ESP. Note that the alternate care-of address option is a part of mobility header that is automatically encrypted by ESP. Mobility header format for BU and BA packets are shown in Fig. 1.

We still have a problem with removing HAs because according to the standard Mobile IPv6, each MN should send a BU message to the HoA of another MN. This packet is received by the HA of the destination MN and is tunneled to that MN as illustrated in Fig. 2. After that, they can communicate using their CoAs without the participation of the HAs. If an MN changes its CoA, it should subsequently send a BU message to the HoA of another MN. To solve this problem and have the process given by Fig. 3, instead of the one given by Fig. 2, each MN should check the Binding Cache when it wants to send a BU message. Therefore, MNs can send BU messages directly to the CoA of the peer without using HAs. We also need to force MNs to check the BUL before sending BA messages in order not to use their HoAs as the source of BA messages.

Other modifications in the standard Mobile IPv6 protocol for the proposed method are presented below:



```
|    8 bits     |    8 bits     |    8 bits     |    8 bits     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Payload Proto | Header Len=03 |  MH Type=05   |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |          Sequence #           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|H|L|K|        Reserved       |           Lifetime            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type = 01(PadN)| Option Len=00 |   Type = 3    |  Length = 16  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                   Alternate Care-of Address                   +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

(a)

```
|    8 bits     |    8 bits     |    8 bits     |    8 bits     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Payload Proto | Header Len=01 |  MH Type=06   |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |    Status     |K|  Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Sequence #           |           Lifetime            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type = 01(PadN)| Option Len=02 |             0000              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

(b)

FIGURE 1. Mobility header format for (a) binding update message and (b) binding acknowledgement message.

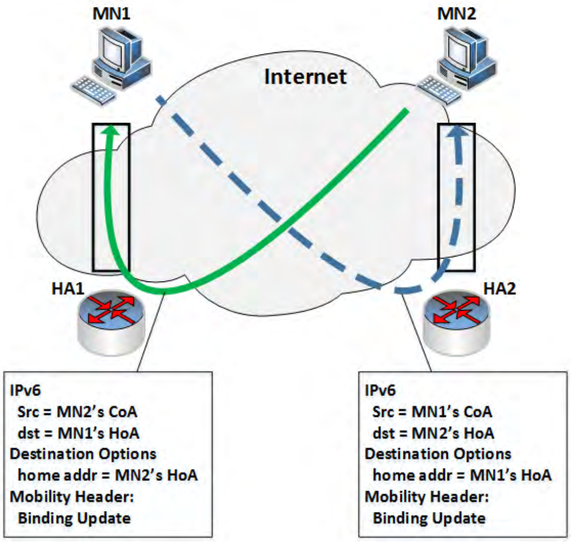


FIGURE 2. Standard Binding Update process between two MNs.

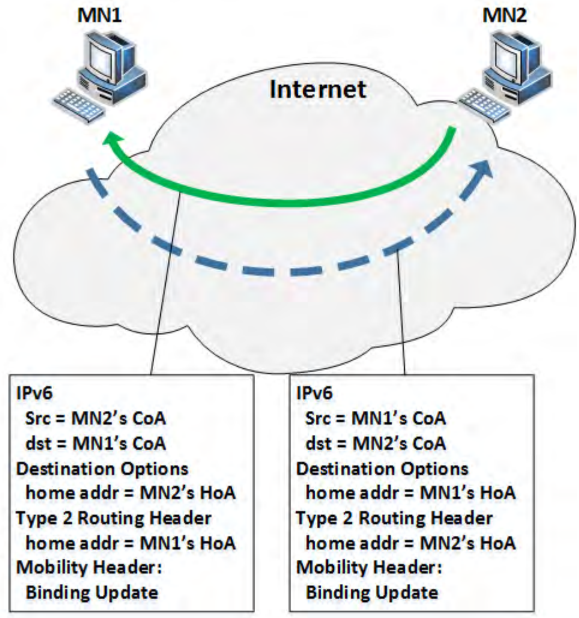


FIGURE 3. Binding Update process using Binding Cache between two MNs.

- IPsec is leveraged for encryption and as a proof of HoA ownership for SRO process. In this way, receiving a BU message protected by IPsec is a proof of HoA ownership. Note that when IPsec is used between two peers, every packet de facto contains a simple piece of information (Security Parameter Index) that gives access to address information (HoAs) for both peers and the shared key. Through the use of SRO process HAs are not needed, and consequently, the HoAs (permanent IP addresses) are not accessible.
- Each new CoA is created and announced by a BU message before removing the previous CoA. In fact, the previous CoA will be removed after receiving the BA message from the peer. More specifically, each MN constantly generates a new CoA and registers it after checking against current occupancy (via a Neighbor Solicitation message). Then the MN sends a Ping Request packet from the new CoA to the IP address of its home router. In this way, the new CoA (with the MAC address) is stored in the table of the home router (to avoid any delay for the first packet with this CoA as its destination). Following the Mobile IPv6 protocol, the MN sends a BU message, and after receiving the relevant BA message, the MN will remove the previous CoA.

Using the proposed method, remote attacks that need to target specific IP address can be prevented. However, we still have a problem with two other types of attacks: (1) black hole attack as a part of DoS attacks (in which a compromised router on the path between two hosts discards packets instead of forwarding them) and (2) bandwidth depletion DDoS attack (that only needs the subnet ID instead of the exact IPv6 address of a target). To prevent these attacks we need to have more than one path between hosts. For example, consider an MN that has two different physical interfaces, one connected to a Cellular link and the other to an Ethernet link. In turn, the MN is connected to two routers that act as foreign networks in Mobile IPv6. In this case, the MN can quickly switch between these two links. For this purpose, the MN needs to register its next CoA on the second link and updates the peer with this new CoA. Therefore, the subsequent data packets will be sent and received through the second link. Hence, the handover delay because of changing links is zero. As a suggestion, a keepalive signal can be used between two MNs to check that the link between them is operating. If the link is broken or the link delay is more than a threshold, each peer can switch between their routers. As another suggestion, the second link can always be used as a redundant path.

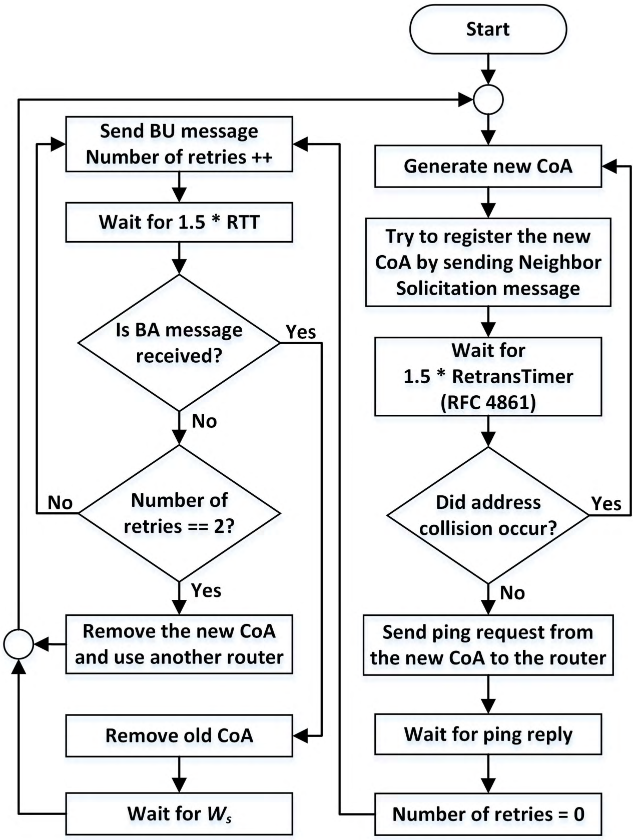

Start
Send BU message
Number of retries ++
Wait for 1.5 * RTT
Is BA message received?
Yes
No
No
Number of retries == 2?
Yes
Remove the new CoA and use another router
Remove old CoA
Wait for W_s
Generate new CoA
Try to register the new CoA by sending Neighbor Solicitation message
Wait for 1.5 * RetransTimer (RFC 4861)
Did address collision occur?
Yes
No
Send ping request from the new CoA to the router
Wait for ping reply
Number of retries = 0

FIGURE 4. IP address rotator flowchart.

The IPv6 address prefixes of routers of each path should be stored as different preferences to implement additional paths. When switching between paths is needed, each host needs to switch between IPv6 address prefix preferences. Note that in our modification the old CoA should still be accessible until receiving the BA message from the peer host for the new CoA. Therefore, the probability of packet dropping due to the change of paths (subnets) is zero.

The flowchart of IP address rotator is shown in Fig. 4. A new generated CoA should be tested to make sure that it is a free IP address by sending a Neighbor Solicitation message. The default waiting time to detect an address collision can be calculated as 1.5 times the value of RetransTimer (default: 1 second), times the value of DupAddrDetectTransmits (default: 1). RetransTimer and DupAddrDetectTransmits are specified in Neighbor Discovery for IP version 6 [39] and Stateless Address Autoconfiguration [34], respectively. So, if a Neighbor Advertisement message is not received within 1.5 seconds, the CoA will be registered. To remove any delay, the new CoA will be added to the router table by sending a Ping Request message from this new CoA to the IP address of the router. When the router receives the ping request, it will send a Neighbor Solicitation message because it does not have the new CoA in its table. Upon receiving the Neighbor Advertisement message from the MN, the router will send back a Ping Reply message. Now the new CoA is registered and ready to use. After that, a BU message is sent to the peer host. If the BA message is not received (as the confirmation of the BU message) after two retries (number of retries depends on the network parameters), a new CoA should be created on the next router (if exists). Recall that the previous CoA is removed after receiving BA message from the peer host.

W_s in the flowchart is the waiting time for the next shuffling period. The fastest shuffling interval can occur if we select zero for W_s. Note that the length of W_s is a network parameter that can be selected independently and dynamically on each peer host. The shorter W_s, the more signaling overhead but, the more resilient to attacks. As a suggestion, a longer W_s can be selected as the default value and if an attack or suspicious activity is detected by anomaly-based or signature-based detection strategies, W_s should be decreased. The minimum shuffling interval is suggested in this study to be calculated by the following equation.

$$(Pc + 1) \times (Tc + 1.5s + 2 \times RTT(MN_i, Router_i)) + T_e + RTT(MN1, MN2) \qquad (1)$$

where $RTT(MN1, MN2)$ equals the mean round-trip time between MN1 and MN2. T_c is the mean calculation time for generating a random IP address and creating packets. T_e is the time needed for encrypting a BU packet by IPsec. P_c is the probability of address collision. To estimate the minimum shuffling interval, we should note that P_c is very small because of the huge address space of IPv6. Furthermore, T_c and T_e are negligible in comparison with the network delay. Therefore, according to (1), if $RTT(MN1, MN2) \gg RTT(MN_i, Router_i)$, then, $1.5s + RTT(MN1, MN2)$ dominates the minimum shuffling interval.

The final scheme of the proposed method is shown in Fig. 5. Recall that this method is a combination of Mobile IPv6, IPsec with IKE_v2, and multiple CoAs registration. We next explain the initialization steps, security, and privacy of the proposed method.

A. INITIALIZATION STEPS

In this subsection, we present the initialization steps to start the communication between peer hosts (MNs). In the first step, the HoA of each MN should have a prefix different from that of the MN's subnet received by Route Advertisement messages. As such, the MNs will see themselves sitting in foreign networks and will subsequently register default CoAs in these networks. In the next step, a manual exchange of configuration information is needed that includes a default CoA of the peer and pre-shared key. Default CoAs are used to set the first entries of the Binding Cache and the BUL in both MNs. The pre-shared key is used for IKE_AUTH Exchange of IKE_v2. Note that other methods of authentication can also be used for IKE_v2 instead of the pre-shared key. These methods of authentication include RSA certificates, elliptic curve digital signature algorithm certificates, and extensible authentication protocol. After these two steps, the MNs can start the IP address rotator to change their CoAs.

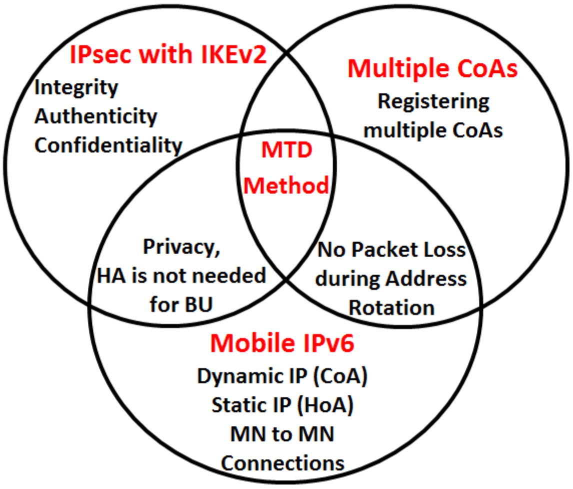


FIGURE 5. **Final scheme of the proposed method.**

If one of the peers (MN1) reboots, another peer (MN2) should use its default CoA and wait for the first packet from MN1. Then both of them can start the IP address rotator. The reboot of a peer can be detected by not receiving BA messages or the keepalive signal (if any) through all existing paths.

B. SECURITY OF MOBILE IPv6

Here we explain some possible attacks against the standard Mobile IPv6 route optimization mechanism [38]. Furthermore, we compare protection solutions used by Mobile IPv6 and the proposed method.

First, if the route optimization mechanism was not authenticated, an attacker could send spoofed BU messages from anywhere on the Internet. As a result, the attacker could redirect all packets between the MN and CNs to itself (attack against secrecy and integrity) or an arbitrary IP address (flooding attack). In Mobile IPv6, these types of attacks are not possible due to the use of authentication in the return routability procedure. The proposed method is also resistant to these types of attacks because of using SRO that leverages

IPsec for authenticating BU messages. Recall that ESP header encrypts the alternate care-of address option of each BU message. In this way, a receiver can authenticate each BU message by comparing the source IP address in the packet header with the encrypted IP address in the alternate care-of address option.

Second, one of the most critical attacks against the standard Mobile IPv6 is the *inducing of unnecessary binding updates*. Unfortunately, the use of authenticated BU messages cannot prevent this type of attack. The impact of the attack becomes more severe as more resources are consumed by the route optimization mechanism. According to default parameters of Mobile IPv6 protocol, when an MN receives a packet from a new CN via its HA, the MN should start the return routability procedure. This procedure is initiated by creating a new entry in the BUL and sending two packets to the CN (the Home Test Init and Care-of Test Init). These two packets will be retransmitted if the MN does not receive Home Test and Care-of Test packets from the CN after a retransmission interval. The retransmission interval is based on an exponential back-off process in which the initial retransmission timer is set to INITIAL_BINDACK_TIMEOUT (default: 1 second) and is doubled upon each retransmission until the time-out period reaches the value MAX_BINDACK_TIMEOUT (default: 32 seconds). This process is finished after 210 seconds as the default lifetime of the BUL entry (MAX _TOKEN_LIFETIME) [12].

An attacker can exploit the retransmission process by sending a spoofed packet to the HoA of an MN. The spoofed packet should look like as if it comes from a new CN. This packet is tunneled to the MN via its HA. Once the MN receives the packet, it starts the route optimization mechanism with this fake CN. As a result, the MN repeats sending two packets eleven times and subsequently removes the entry from its BUL. Therefore, if an attacker induces an MN to initiate the route optimization mechanism with a non-existent CN, the MN will send 22 packets while keeping the corresponding entry in the table for 210 seconds. In practice, the attacker would trigger the MN to initiate a large number of route optimizations with fake CNs.

We simulated this attack against the standard Mobile IPv6. For this purpose, we used Scapy [40], a powerful interactive packet manipulation program, to send some Ping Request packets with a random IP address in the source of the packet. The prefix of these random addresses was the same as the attacker's address prefix to avoid ingress filtering. In this test, we created a flooding attack for 32 seconds with 12,800 Ping Request packets to the HoA of an MN. In our experiments, the MN sent a Ping Reply packet per each received request packet and also sent 281,600 (12,800×22) packets during the retransmission process of the return routability procedure. Therefore, forcing the MN to send 294,400 $(281,600 + 12,800$ Ping Reply) packets during 242 seconds was the result of this attack.

However, it is not possible to launch this attack against the proposed method because we do not use any HA and accordingly, peers are not accessible by their HoAs. Hence, even though the Mobile IPv6 is used, the above vulnerability is not inherited by the proposed method. This is a significant result in terms of security of the method.

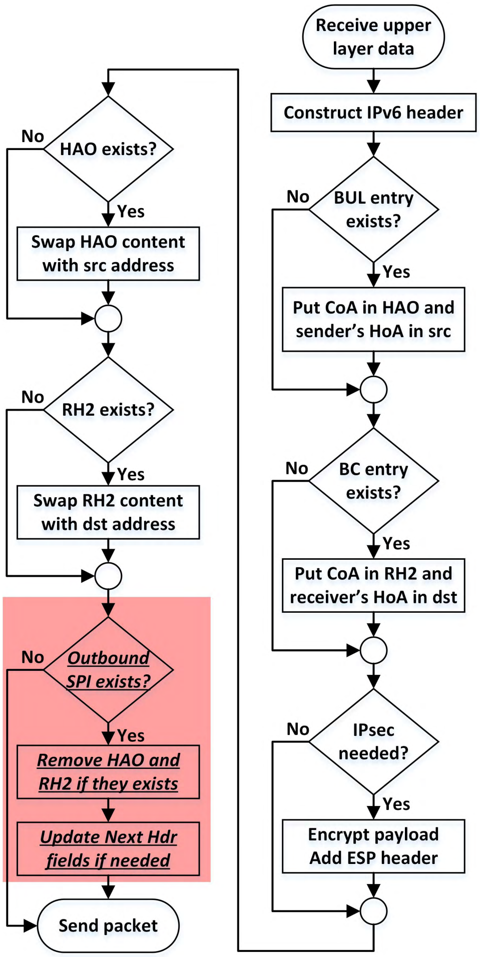

Receive upper layer data
Construct IPv6 header
BUL entry exists?
No
Yes
Put CoA in HAO and sender's HoA in src
BC entry exists?
No
Yes
Put CoA in RH2 and receiver's HoA in dst
IPsec needed?
No
Yes
Encrypt payload Add ESP header
HAO exists?
No
Yes
Swap HAO content with src address
RH2 exists?
No
Yes
Swap RH2 content with dst address
Outbound SPI exists?
No
Yes
Remove HAO and RH2 if they exists
Update Next Hdr fields if needed
Send packet

FIGURE 6. Sending packets processing flowchart.

C. PRIVACY IMPROVEMENT VIA IPsec

When an MN (MN1) wants to send a data packet to another MN (MN2), it will have already supplied the HoAs as the source and destination addresses in the packet header. Next, MN1 checks the BUL to see if it has already sent a BU message to MN2. If it is found, MN1 includes its CoA in the home address option. MN1 then checks its binding cache to see if MN2 has sent a BU message to MN1. If it is found, MN1 constructs a routing header type 2 and places the CoA of MN2 inside this header. The packet with HoAs in the source and destination addresses of the header reaches IPsec. After encrypting and adding headers, the home address option is swapped with the source address and the routing header type 2 is swapped with the destination address of the packet header. When the packet is received by MN2, the headers are processed in the order that they appear in the packet. Therefore, HoAs are inserted in the source and destination

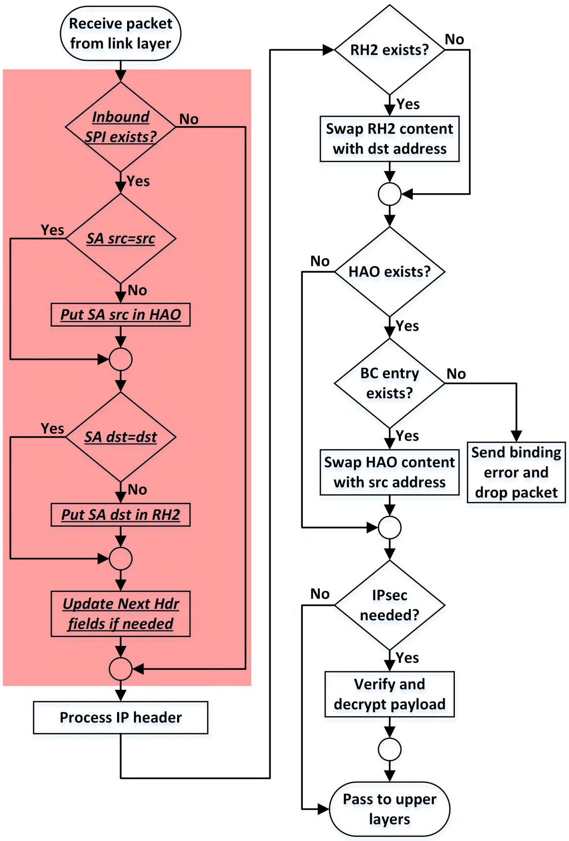


FIGURE 7. Received packets processing flowchart.

addresses of the packet header before the packet reaches IPsec. In this way, the IPsec implementation always sees HoAs in the source and destination addresses of the packet header. Note that it is the best to use HoAs as the selectors in IPsec to avoid changing the Security Association (SA) every time a new CoA is defined [41, pp. 116–119].

According to the standard implementation of Mobile IPv6 and IPsec, when a packet is on the path, the source and destination addresses in the header are CoAs of MN1 and MN2. However, IPsec does not encrypt the routing header type 2 and the home address option that show HoAs of the source and destination. To improve the privacy, we can resolve this problem by removing the destination option header (and the routing header type 2) from all packets. Note that the Security Parameter Index (SPI) found in the ESP header is sufficient to get access to the HoAs (the real source/destination of a packet). The highlighted parts in the flowcharts in Fig. 6 and Fig. 7 show our modification in the standard MN operation with IPsec (HAO and RH2 in these flowcharts are the home address option and the routing header type 2, respectively). The packet format before and after removing the destination option header/the routing header type 2 are shown in Fig. 8.

We propose the use of IKEv2 as a standard method for key management. IKEv2 can improve the security of IPsec and prevent replay attacks. Some negotiations should be done between peers to define SPIs (a total of four messages) to start IKEv2. After that, we can remove HoAs in the source peer and retrieve them via SPIs in the destination peer. Therefore, the destination option header/the routing header type 2 are only needed for the first four messages. Recall that a pre-shared key can be used for IKE_AUTH Exchange that is the first step of IKEv2. The details are out of the scope of this paper.

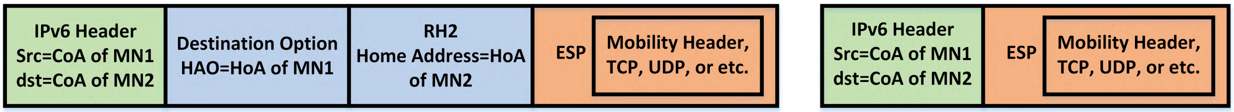
VI. IMPLEMENTATION RESULTS

A proof of concept prototype implementation of the proposed method is developed to prove the validity and evaluate the performance of the design. In our prototype, we used OpenPLC [42], an open source PLC runs on a RaspberryPi mini-computer running Raspbian Jessie. OpenPLC is networked through a set of routers (using Modbus/TCP) to an open source HMI software (ScadaBR [43]) installed on a desktop computer containing 2.4GHz Intel Core 2 Duo CPU with 4GB DDR2 800MHz RAM running Ubuntu 14.04. Linux kernel version 3.8.2 with enabled mobility options is compiled and installed on both devices. An open source implementation of Mobile IPv6 (UMIP) for Linux is used. Some changes have been done on the original UMIP to support the proposed method. Both devices act like MNs of Mobile IPv6. Router R1 is used as the heart of the Internet. The WAN ports of other routers (R2, R3, and R4) are connected to the LAN ports of R1. To test the prevention of bandwidth depletion DDoS attacks, we used an extra path between OpenPLC and R1. OpenPLC is connected to both R3 and R4 via two network interfaces. We used a USB network adapter to add an extra network interface to OpenPLC. In the configuration file of UMIP (on OpenPLC), stored in /usr/local/etc/mip6d.conf, we listed both interfaces with two preference numbers. The preference number of the interface connected to R3 is the smallest one (highest priority). Therefore, OpenPLC uses the path through R3 as the default path to R1.

The prefix of each HoA is different from the prefix received by Route Advertisement messages. So both devices (MNs) see themselves sitting in foreign networks and subsequently register CoAs in these networks. The network topology of the testbed is illustrated in Fig. 9. A function is added to the source code of UMIP for changing CoAs of the MNs every 10 seconds. We compared this new version of MTM6D (MTM6D II) with the previous version of MTM6D [8] and MT6D method [29]. Having zero packet loss in MTM6D II is the best advantage in comparison with MTM6D as explained in Section VI-B.

A. OVERHEAD

There are two types of overheads in MTM6D II. First, there is some signaling overhead; some extra packets are used due to the CoA notification process (binding update procedure), which involves the BU and the BA messages. The ACK bit of each BU message forces the peer host to send back a BA message as a confirmation. This process ensures that the peer host receives the BU message and updates its binding cache



(a)

(b)

FIGURE 8. Packet format before (a) and after (b) removing the destination option header and the routing header type 2.

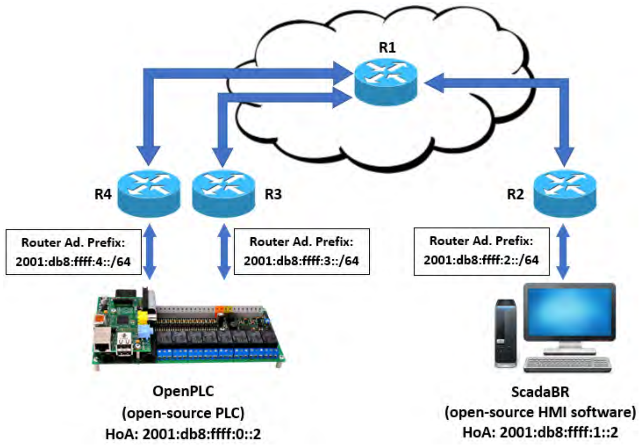


FIGURE 9. The network topology of the testbed.

entry before removing the previous CoA by the sender. Each round of changing IP address needs two message transmissions at each MN (BU and BA messages). As presented in Section V, we do not use any HA in the proposed method in order to prevent access to HoAs, i.e., the permanent IP addresses of peers through the Internet. Therefore, SRO is used as the route optimization method that does not need the participation of any HA. Considering the mobility header format for BU and BA messages (shown in Fig. 1) and the use of SPI of ESP header for removing the destination option header and the routing header type 2 (presented in Section V-C), we can calculate the size of BU/BA packets:

$$BU : 14B(Ethernetheader) + 40B(IPv6header) + 24B(IPsec(ESP)) + 32B(Mobilityheader) = 110B. \tag{2}$$

$$BA : 14B(Ethernetheader) + 40B(IPv6header) + 24B(IPsec(ESP)) + 16B(Mobilityheader) = 94B. \tag{3}$$

Therefore, the total signaling overhead to update a peer host with a new CoA is 204 bytes. For example, if each host changes its CoA every 10 seconds, the signaling overhead is $40.8B/sec$. In the standard implementation of Mobile IPv6, the BU and BA messages are 110 bytes each. Note that instead of IPsec header, they have the routing header type 2 and the home address option in each signaling packet. However, in the standard Mobile IPv6, there are also four extra messages due to the use of the return routability procedure. Therefore, the overhead of route optimization in the original Mobile IPv6 is equal to 660 bytes (in comparison with 204 bytes in the proposed method).

Second, there is transmission overhead; each data packet transmission between hosts has some overhead. For each data packet, we have 24 extra bytes of overhead due to the use of IPsec with ESP protocol. Note, however, that the additional bytes that come from the use of IPsec are not caused by the proposed method. Any secure communication utilizing IPsec ESP would incur the same overhead. On the other hand, MT6D encapsulates each packet using UDP to hide the original IP addresses and uses virtual IP addresses. The overhead of MT6D equals 62 bytes [29]. In MT6D encapsulation, the Ethernet header is also overwritten to anonymize the MAC addresses. However, the proposed method does not have any MAC address anonymity because the packet source MAC address will be changed automatically when it is received by the first router. For a fair comparison, we assume only 48 bytes overhead (IPv6 header and UDP header) for MT6D.

To compare the overhead of MT6D and the proposed method, different shuffling intervals (t) are used. Let O_i be the overhead per each packet for method i and N be the mean number of packets per second, then we have:

$$O_{MTM6D\,II} = 24B + \left(\frac{408B}{N \times t}\right). \tag{4}$$

Recall that $O_{MT6D} = 48B$. So $O_{MT6D} < O_{MTM6D\,II}$ if $(N \times t) < 17$.

This comparison is shown in Fig. 10. We used three different shuffling intervals for MTM6D II in this comparison. Two seconds is used as an approximation to the minimum value. Furthermore, 10 seconds is used as the default value in MT6D. One minute is also used as an example for a long shuffling interval. According to these calculations, when the shuffling interval equals 10 seconds (as an example) and the mean number of packets per second is greater than 1.7, the overhead per packet of the proposed method is less than MT6D.

B. HANDOFF DELAY

In MTM6D and MTM6D II when an MN changes its CoA, it should update the peer with the new CoA. MTM6D has a small window of vulnerability during the handoff delay because packets sent by the peer (addressed to now-defunct CoA) could not be delivered. However, in MTM6D II, we use multiple CoAs such that the old CoA is kept alive until the

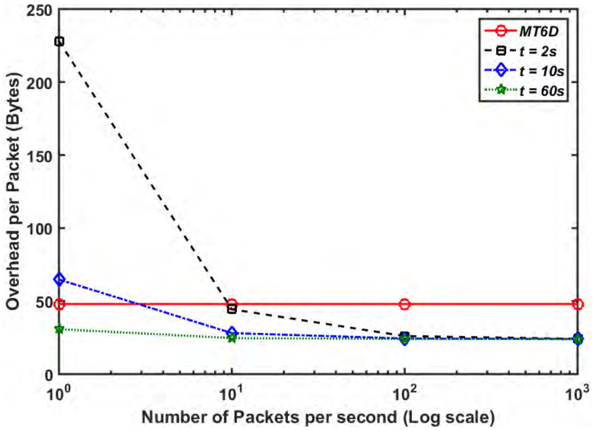


FIGURE 10. Overhead per packet as a function of number of packets per second.

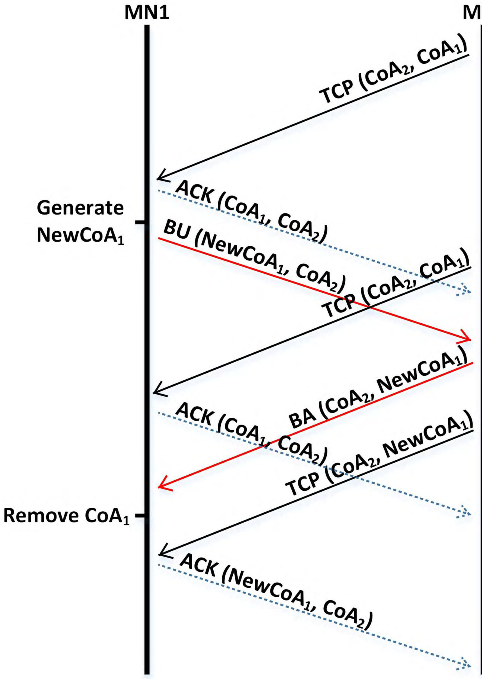


FIGURE 11. Zero packet loss during the handoff delay.

new CoA has been received by the peer. The old CoA is removed once the peer node sends back the BA message showing that the new CoA is saved in the peer. Recall that the handoff delay equals the round-trip time between peers (connection latency times two). Fig. 11 illustrates this process. In this figure, CoA_i is the current CoA of MNi. One IP address rotation of MN1 is shown in the figure.

In MT6D each host maintains three addresses: the previous interval address, the current interval address, and the next interval address to eliminate any packet loss during handoff delay. To better understand the effect of handoff delay on the communication, we compared MTM6D II with MTM6D in a high latency scenario. We assumed 600 ms as the round-trip time in our implementations.

TABLE 1. UDP packet loss rate.

Number of packets per second	Packet Loss Rate	
	MTM6D	MTM6D II
10	5.80%	0%
100	5.85%	0%
1000	5.77%	0%

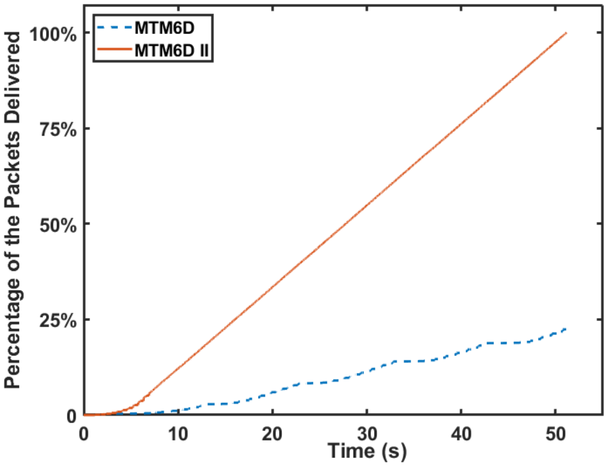


FIGURE 12. Percentage of TCP packets delivered over time in MTM6D and MTM6D II.

1) UDP TEST

For this test, we selected one MN to transmit UDP packets to another MN. As mentioned earlier, the time duration for updating the peer equals 600 ms. We used 10 seconds as the shuffling interval for both MTM6D II and MTM6D. Therefore, the handoff delay ratio equals $(0.600/10) = 6\%$. In MTM6D, all data packets are lost during the handoff, but, in MTM6D II the packet loss rate equals zero. Table 1 shows the experimental results for various numbers of UDP packets per second generated by a traffic generator.

2) TCP TEST

For TCP test, one MN is utilized to send 1000 TCP packets per second (each 500B) to another MN. Shuffling interval equals 10 seconds. In MTM6D, the handoff delay has an important effect on the throughput. During the handoff delay, TCP experiences timeout and resends the unacknowledged packets and goes to slow start. However, in the proposed method we do not have any packet loss. As shown in Fig. 12, the ratio of delivered TCP packets to the sent packets equals 22.96% for MTM6D (in comparison with 100% in MTM6D II).

C. BANDWIDTH DEPLETION DDoS ATTACKS PREVENTION TEST

As explained in Section V, remote attacks that need to know their intended target's IP address can be prevented by rotating

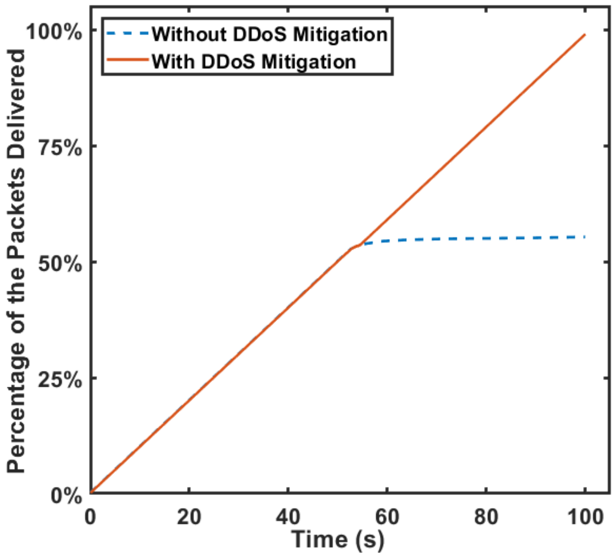


FIGURE 13. Percentage of TCP packets delivered over time with and without DDoS mitigation method.

TABLE 2. Comparison between MTM6D II and MT6D.

	MTM6D II	MT6D
Mobility support	✓	✗
Independent IP address rotation interval	✓	✗
Black hole and bandwidth depletion DDoS attacks resistance[a]	✓[b]	✗
Independent encryption, authentication method, and key-size	✓[c]	✗
Zero packet loss probability	✓	✗[d]
No extra requirements	✓	✗[e]
Signaling overhead per each host per each rotation interval	$220B$	$0B$
Transmission overhead per each data packet	$24B$	$62B$

[a] Capability of changing links (if any) without connection disruption.

[b] If a host is connected to two links (as an example) and both of them are under attack then MTM6D II also cannot prevent these attacks.

[c] Different algorithms and key-size can be used based on IPsec with IKE_v2.

[d] There is a possibility of packet loss due to address collision.

[e] Relatively tight time synchronization is needed.

the IP address. However, not all remote attacks need to know the exact IP address of their target(s). In fact, remote attacks could occur against routers on the path between two connected nodes. Using IPsec with IKE_v2 adds confidentiality, integrity, and replay attacks protection. Therefore, attackers will only be able to increase the network delay by bandwidth depletion DDoS attacks. Because of the mobility support feature of MTM6D II, a node can switch between multiple paths (if exists) if it detects an extra network delay without disrupting TCP sessions or dropping data packets.

For this test, we connected a switch (as an input) and a lamp (as an output) to OpenPLC. A simple PLC program is uploaded to OpenPLC to turn on and off the lamp based on the status of the switch. The HMI (installed on the desktop computer) sends a query message (every 200 ms) to OpenPLC to read the status of the switch. As the response, OpenPLC sends back a response message to the HMI. The default round-trip time between the peers in this test is 100 ms. To simulate a bandwidth depletion DDoS attack, we manually (via a script code) increased the network delay on R3 (the default router of OpenPLC). In fact, from 50 seconds after the start of this test, the network delay is doubled per second. Therefore, the round-trip time between the peers starts from 100 ms and increases to 200 ms at the 50th second, 400 ms at the 51st second, 800 ms at the 52nd second and so forth.

Two scenarios are considered: (1) MTM6D II with DDoS Mitigation and (2) MTM6D II without DDoS Mitigation. In the first scenario, a keepalive signal (ping requests) is used to check that the link between the peers are operating and the network delay is below a threshold (150 ms). OpenPLC sends a ping request packet every second to measure the network delay of the current path. OpenPLC detects a delay above the threshold after the 50th second and decides to use another path through R4. Therefore, OpenPLC registers a new CoA on R4 and sends a BU packet to the HMI and after that OpenPLC switches from R3 to R4. In the second scenario, we did not implement the DDoS mitigation method. The comparison between these two scenarios is shown in Fig. 13. Please note that applying the black hole attack has a similar result of increasing the network delay on R3 to a large number.

Table 2 shows a brief comparison between the proposed MTD method (MTM6D II) and MT6D (the closest method). Significant improvements can be seen in terms of security, availability, flexibility, independence, etc. Interested readers are referred to Sections I and II for more details on the comparison.

D. WORK IN PROGRESS

To test the security enhancements completed on the OpenPLC system a set of cyber-attacks will be developed. We will develop a set of cyber-attacks which match the threat model from Section III. The cyber-attacks will be ported to target our testbed. Existing cyber-attacks are available to scan for and enumerate SCADA systems, inject, alter and replay control and sensor measurement network packets, and to cause a denial of service. Each existing cyber-attack will be reviewed and either ported or replaced with a similar attack targeting the testbed. A metric will be developed to measure the effectiveness of each cyber-attack. We believe that all of these cyber-attacks will not be effective when MTM6D II is implemented.

VII. CONCLUSION

In this paper, we presented a framework for building a secure and private peer to peer communication for SCADA networks with a novel Mobile IPv6 based Moving Target Defense strategy. We showed that our approach, MTM6D II, can help thwart remote cyber-attacks against peer hosts by making the hosts difficult to be found. Towards this purpose, dynamic random IP addresses are used instead of static permanent IP addresses. Furthermore, we utilize RFC 4449 along with IPsec with IKEv2 for creating a secure route optimization (SRO) method without the participation of any HA. Applications of this new route optimization method are not limited to the proposed MTD method but also can be used for other applications of Mobile IPv6. Removing the destination option header (and the routing header type 2) from all packets is proposed to improve privacy and anonymity for communicating hosts and decrease overhead. As another security improvement, using additional paths between the peer hosts with the ability to switch between these paths without any delay or packet loss is proposed to combat black hole and bandwidth depletion DDoS attacks. Use of the combination of standard protocols instead of creating a new protocol made the proposed method to be independent of a specific algorithm or key size for encryption, authentication, and key distribution. This portability feature makes it easy to implement this method for different applications. The end result is a solution that may also be combined with existing defensive measures to form a robust Defense in Depth solution.

Although the results presented in this work claimed to outperform the recent results of the MT6D method, the scalability issue of both methods remains to be investigated. The most significant part of future work is adopting the presented MTD method to support one-to-many and many-to-many communications.

REFERENCES

[1] W. Gao, T. Morris, B. Reaves, and D. Richey, “On SCADA control system command and response injection and intrusion detection,” in *Proc. eCrime Res. Summit*, Oct. 2010, pp. 1–9.

[2] *Slammer Worm Crashed Ohio Nuke Plant Network*. Accessed: Jan. 3, 2018. [Online]. Available: https://www.securityfocus.com/news/6767

[3] *W32_Stuxnet_Dossier.pdf*. Accessed: May 23, 2018. [Online]. Available: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[4] *ICS Focused Malware (Update A) | ICS-CERT*. Accessed: Jan. 3, 2018. [Online]. Available: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-14-176-02A

[5] *Scada Strangelove*. Accessed: Jan. 3, 2018. [Online]. Available: http://scadastrangelove.blogspot.com/

[6] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on SCADA systems,” in *Proc. IEEE Int. Conf. Internet Things, Cyber, Phys. Social Comput.*, Oct. 2011, pp. 380–388.

[7] S. East, J. Butts, M. Papa, and S. Shenoi, “A taxonomy of attacks on the DNP3 protocol,” in *Critical Infrastructure Protection III*, C. Palmer and S. Shenoi, Eds. Berlin, Germany: Springer, 2009, pp. 67–81.

[8] V. Heydari and S.-M. Yoo, “Securing critical infrastructure by moving target defense,” in *Proc. 11th Int. Conf. Cyber Warfare Secur. (ICCWS)*, 2016, pp. 382–390.

[9] V. Heydari, “IP hopping by mobile IPv6,” in *Handbook Cyber-Development, Cyber-Democracy, Cyber-Defense*, E. G. Carayannis, D. F. J. Campbell, and M. P. Efthymiopoulos, Eds. Cham, Switzerland: Springer, 2017, pp. 1–28, doi: 10.1007/978-3-319-06091-0_49-1.

[10] V. Heydari, S. M. Yoo, and S. I. Kim, “Secure VPN using mobile IPv6 based moving target defense,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[11] V. Heydari, “Preventing ssh remote attacks using moving target defense,” in *Proc. 13th Int. Conf. Cyber Warfare Secur. (ICCWS)*, 2018, pp. 272–280.

[12] C. Perkins, D. Johnson, and J. Arkko, *Mobility Support in IPv6*, document RFC 6275, Jul. 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6275.txt

[13] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, *Internet Key Exchange Protocol Version 2 (IKEv2)*, document RFC 7296, Oct. 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7296.txt

[14] J. M. Taylor and H. R. Sharif, “Enhancing integrity of modbus TCP through covert channels,” in *Proc. 11th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2017, pp. 1–6.

[15] E. Pricop, J. Fattahi, N. Parashiv, F. Zamfir, and E. Ghayoula, “Method for authentication of sensors connected on modbus TCP,” in *Proc. 4th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Apr. 2017, pp. 0679–0683.

[16] T. Alves, R. Das, and T. Morris, “Embedding encryption and machine learning intrusion prevention systems on programmable logic controllers,” *IEEE Embedded Syst. Lett.*, to be published, doi: 10.1109/LES.2018.2823906.

[17] T. Alves, T. Morris, and S.-M. Yoo, “Securing SCADA applications using openPLC with end-to-end encryption,” in *Proc. 3rd Annu. Ind. Control Syst. Secur. Workshop (ICSS)*. New York, NY, USA: ACM, 2017, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/3174776.3174777

[18] P. Wood, C. Gutierrez, and S. Bagchi, “Denial of service elusion (DoSE): Keeping clients connected for less,” in *Proc. IEEE 34th Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2015, pp. 94–103.

[19] H. Wang, Q. Jia, D. Fleck, W. Powell, F. Li, and A. Stavrou, “A moving target DDoS defense mechanism,” *Comput. Commun.*, vol. 46, pp. 10–21, Jun. 2014.

[20] S. Venkatesan, M. Albanese, K. Amin, S. Jajodia, and M. Wright, “A moving target defense approach to mitigate DDoS attacks against proxy-based architectures,” in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2016, pp. 198–206.

[21] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, “Catch me if you can: A cloud-enabled DDoS defense,” in *Proc. IEEE/IFIP Dependable Syst. Netw.*, Jun. 2014, pp. 264–275.

[22] B. Danev, R. J. Masti, G. O. Karame, and S. Capkun, “Enabling secure VM-vTPM migration in private clouds,” in *Proc. 27th Annu. Comput. Secur. Appl. Conf. (ACSAC)*. New York, NY, USA: ACM, 2011, pp. 187–196. [Online]. Available: http://doi.acm.org/10.1145/2076732.2076759

[23] Y. Zhang, M. Li, K. Bai, M. Yu, and W. Zang, “Incentive compatible moving target defense against VM-colocation attacks in clouds,” in *Information Security and Privacy Research*. Berlin, Germany: Springer, 2012, pp. 388–399, doi: 10.1007/978-3-642-30436-1_32.

[24] H. Okhravi, A. Comella, E. Robinson, and J. Haines, “Creating a cyber moving target for critical infrastructure applications using platform diversity,” *Int. J. Critical Infrastruct. Protection*, vol. 5, no. 1, pp. 30–39, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874548212000030

[25] J. B. Hong and D. S. Kim, “Assessing the effectiveness of moving target defenses using security models,” *IEEE Trans. Depend. Sec. Comput.* , vol. 13, no. 2, pp. 163–177, Mar./Apr. 2016.

[26] T. M. Gil and M. Poletto, “Multops: A data-structure for bandwidth attack detection,” in *Proc. 10th Conf. USENIX Secur. Symp. (SSYM)*, vol. 10. Berkeley, CA, USA: USENIX Association, 2001, Art. no. 3. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251327.1251330

[27] A. Hussain, J. Heidemann, and C. Papadopoulos, “A framework for classifying denial of service attacks,” in *Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun. (SIGCOMM)*. New York, NY, USA: ACM, 2003, pp. 99–110. [Online]. Available: http://doi.acm.org/10.1145/863955.863968

[28] J. H. Jafarian, E. Al-Shaer, and Q. Duan, “Openflow random host mutation: Transparent moving target defense using software defined networking,” in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*. New York, NY, USA: ACM, 2012, pp. 127–132. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342467

[29] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront, “MT6D: A moving target IPv6 defense,” in *Proc. AFCEA/IEEE MILCOM*, Nov. 2011, pp. 1321–1326.

[30] *Modbus_Application_Protocol_V1_1b3.pdf*. Accessed: May 23, 2018. [Online]. Available: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

[31] *IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)*, Standard 1815-2012, Oct. 2012, pp. 1–821.

[32] *Profinet System Description*. Accessed: May 23, 2018. [Online]. Available: http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/automaatiotekniikka/teollinen_tiedonsiirto/profinet/man_pnsystem_description.pdf

[33] *Ethernet/IP Quick Start for Vendors Handbook*. Accessed: May 23, 2018. [Online]. Available: https://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00213R0_EtherNetIP_Developers_Guide.pdf

[34] S. Thomson, T. Narten, and T. Jinmei, *IPv6 Stateless Address Autoconfiguration*, document RFC 4862, Sep. 2007. Accessed: May 23, 2018. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4862.txt

[35] D. Johnson, C. Perkins, and J. Arkko, *Mobility Support in IPv6*, document RFC 3775, Jun. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3775.txt

[36] *Cisco ASA Software IKEv1 and IKEv2 Buffer Overflow Vulnerability*. Accessed: Aug. 5, 2016. [Online]. Available: http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20160210-asa-ike

[37] C. Perkins, *Securing Mobile IPv6 Route Optimization Using a Static Shared Key*, document RFC 4449, Jun. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4449.txt

[38] P. Nikander, J. Arkko, T. Aura, G. Montenegro, and E. Nordmark, *Mobile IP Version 6 Route Optimization Security Design Background*, document RFC 4225, Dec. 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4225.txt

[39] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, *Neighbor Discovery for IP Version 6 (IPv6)*, RFC 4861, Sep. 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4861.txt

[40] Scapy. Accessed: Oct. 4, 2015. [Online]. Available: http://www.secdev.org/projects/scapy/

[41] H. Soliman, *Mobile IPv6*. Boston, MA, USA: Addison-Wesley, 2004.

[42] T. R. Alves, M. Buratto, F. M. de Souza, and T. V. Rodrigues, “OpenPLC: An open source alternative to automation,” in *Proc. IEEE Global Humanitarian Technol. Conf. (GHTC)*, Oct. 2014, pp. 585–589.

[43] *ScadaBR/Wiki/Home*. Accessed: May 23, 2018. [Online]. Available: https://sourceforge.net/p/scadabr/wiki/Home/

VAHID HEYDARI received the M.S. degree in cybersecurity and the Ph.D. degree in electrical and computer engineering from the University of Alabama in Huntsville, Huntsville, AL, USA, in 2016 and 2017, respectively. He is currently an Assistant Professor of computer science and the Director of the Center for Cybersecurity Education and Research, Rowan University, Glassboro, NJ, USA. His research interests include moving target defenses, mobile ad-hoc, sensor, and vehicular networks security. He is a member of ACM, the IEEE Computer Society, and Communications Society.