IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 19, 2020, accepted October 7, 2020, date of publication October 27, 2020, date of current version November 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034226

Hybrid Multilayer Network Traceback to the Real Sources of Attack Devices

MING-HOUR YANG[1], (Member, IEEE), JIA-NING LUO[2], (Member, IEEE), M. VIJAYALAKSHMI[3], AND S. MERCY SHALINIE[3], (Senior Member, IEEE)

[1]Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan City 320314, Taiwan
[2]Department of Information and Telecommunications, Ming Chuan University, Taoyuan City 333321, Taiwan
[3]Network Laboratory, Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai 625015, India

Corresponding author: M. Vijayalakshmi (mviji@tce.edu)

This work was supported in part by the Taiwan Information Security Center (TWISC), and in part by the Ministry of Science and Technology (MOST) under Grant MOST 109-2218-E-009-010, Grant MOST 109-2221-E-130-006, and Grant MOST 109-2218-E-011-007.

ABSTRACT With the advent of the Internet of Things (IoT), there are also major information security risks hidden behind them. There are major information security risks hidden behind them. Attackers can conceal their actual attack locations by spoofing IP addresses to attack IoT devices, law enforcement cannot easily track them. Therefore, a method to trace stealth attacks is required. Conventional IP traceback methods that traceback only attackers on the network layer and cannot infer the path information of a packet traversing the switch. This article proposes a method to simultaneously traceback attack sources at the network layer and the data link layer with only one single packet. Even if the core network contains a switch or if multiple attackers launch attacks from different locations, the method can correctly traceback the true devices responsible for the attacks, and its achievements include a zero false negative rate and a low false positive rate.

INDEX TERMS IP traceback, DDoS attack, attack mitigation, layer 2 traceback, autonomous system, attack detection, IP spoofing, advanced persistent threats.

I. INTRODUCTION

Many manufacturers have connected applications required in our daily lives to the Internet. Using the cloud to centralize storage and analysis systems, they provide various monitoring and management services to render our lives more comfortable and convenient. However, if a system design is not robust or if consumer habits are poor, threats to information security arise. In particular, attacks on equipment related to security and privacy such as automobile driving control, electronic door locks, and Internet of things (IoT) devices security monitors can have disastrous consequences [1], [2].

Incidents of cyberattacks have increased, both in terms of number and scale, and damage time and effects have also intensified. Due to the anonymity of the Internet, cybercrime is difficult to detect, especially for the common distributed denial of service (DDoS) for IoT systems. Moreover, the major challenges remain in dealing with a DDoS attack is to differentiate between normal and malicious packets [3]. The attacker can easily conceal or falsify the true source of the attack using technologies such as a proxy, VPN, fake IP, public network or wireless network, or zombie computers, thereby becoming difficult to trace [4]. This has caused the present-day frequent occurrence of cyberattacks and the continuous emergence of cybercrime, especially the advanced persistent threats (APTs) attacks [5]. However, even an APT attack was detected, to effectively curb cybercrime, the development of packet analysis that easily traces the source of an anonymous attack is a key priority for the present-day development of information security and network forensics [6].

Current methods of tracing back anonymous attacks primarily comprise methods such as packet marking, packet logging traceback, and hybrid IP traceback. The packet-marking method can be divided, according to the frequency of packet marking, into the following: deterministic packet marking, which marks all packets passing through the ingress router of a network [7]–[13], and probabilistic packet marking, which marks passing packets probabilistically [14]–[22]. Furthermore, Liu *et al.* [23] proposed a trust-aware probability marking traceback scheme. The marking rate in is depends

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

on trustable nodes or not. To trace back an attack using a single packet, research has proposed the packet logging traceback [24]–[27] and hybrid IP traceback [28]–[38].

If the IP traceback employs traceback on the network layer, the true location of the attacker is difficult to trace. Praveena *et al.* [36] proposed a log-based traceback that uses the Unicast Reverse Path Forwarding (RPF) function to detect the source. Ling *et al.* proposed a method to alter the TCP flow control messages server-side switches of the Software-Defined Networking (SDN) networks to trace the source [39]. Li *et al.* [40] developed a log-based IP traceback architecture suitable for partial deployment scenario in ISP level. Thus, Baba and Matsuda [41] used the data link layer to trace back the source of attackers. However, this method traces back only to the edge router closest to the attacker and not to the location of the device that launched the attack.

Accordingly, Hazeyama *et al.* [42] transmitted the following information inside a switch to the edge router: port, network interface identifier, virtual local area network identifier, source Media Access Control (MAC) address, destination MAC address, and packet digests. Once an attack occurs, this information can be used to infer the port and network interface identifier of the attack packet and thereby trace the device location of the attack source. However, because this method stores the information of packets from the upper-layer switch only, once the network architecture of the attack source exists on a switch with more than two layers, locating the true device on the attacking end becomes impossible.

Snow and Park [43] proposed methods for hybrid packet marking and storage that placed information entering a switch, such as the port, switch ID, and packet digests, in the existing packet and then transmitted them to the next switch, repeating until the edge server is reached. Although this method can traceback the true device at the attacking end, implementation is difficult because the packet mark cannot readily be attached to data link layer packets that meet special standard.

Marios *et al.* [44] therefore established a bloom filter at every port as a log table. When a packet enters a switch through a port, that switch uses a hash function to obtain an index value and set the index of the bloom filter for that port to 1. After an attack, the index must only be individually confirmed to be 1 for the true device on the attacking end to be identified. However, this method is characterized by the disadvantages of large storage space and a false positive rate that increases substantially with time and the number of packets.

Internet service providers often employ an Internet Exchange Point (IXP) to increase transmission efficiency and lower costs. As a result, more switches exist in the core network environment, which prevents the ports between routers from being in a one-to-one relationship. However, most existing tracebacks fail to consider that the core network environment may include switches [45]. Therefore, when a path is reconstructed using the IP traceback method, tracing back to the actual attack source may be impossible due to a one-to-many situation.

Currently, no attack source traceback method can simultaneously perform packet traceback at the network layer and the data link layer. Therefore, directly tracing the actual attack launch device of the hidden source from the victim end is inefficient. Therefore, this study proposes a method to simultaneously trace attack sources at the network layer and data link layer. This method combines tracebacks at the network layer and data link layer and, in the switch, uses a switch port mirroring device (TAP) for logging, and uses the Time to live (TTL) value of the IP headers as a judgment for terminating the traceback, thereby obtaining the number of routers from the attack source. Even if the core network includes switches, the attack source can be accurately traced using a single packet. The primary contributions of this study are as follows:

- Hybrid network layer and data link layer traceback method.
- Single packet traceback.
- Ability to simultaneously trace several attacks from various sources.
- Inclusion of switches in the core network environment does not decrease the accuracy of attack source tracing.
- Ability to traceback to the true attack source device rather than only to an edge router.
- Zero false negative rate and low false positive rate in tracebacks at the network layer.

Section 2 of this article first defines the attacker's attack model and the environments in which this method is applicable and then details the marking methods and logging mechanisms of the researchers as well as path reconstruction methods following attacks. Next, Section 3 analyzes storage capacity and accuracy and conducts comparisons with other relevant studies. Section 4 introduces the conclusion of this article.

II. HYBRID SINGLE-PACKET TRACING FOR TRUE SOURCE MARKING AND LOGGING

In this article, a single packet traceback is proposed. In comparison to conventional IP tracebacks, this method traces the edge router in front of the attacker and also the true source of the attack or the device that launched the attack. When tracing the true source of the attack, in addition to tracing the attacker's edge router at the network layer, this method further combines tracing technology at the link layer to find the actual source device from which the attacker launched the attack.

In the Internet's routing architecture, the routers of each Internet service provider (ISP) operator will form an autonomous system (AS). An autonomous system is a collection of connected routers on behalf of a single administrative domain that presents a common routing policy. ISPs will exchange packets through the IXP framework, as shown in Figure 1, the core network includes switches, such that one

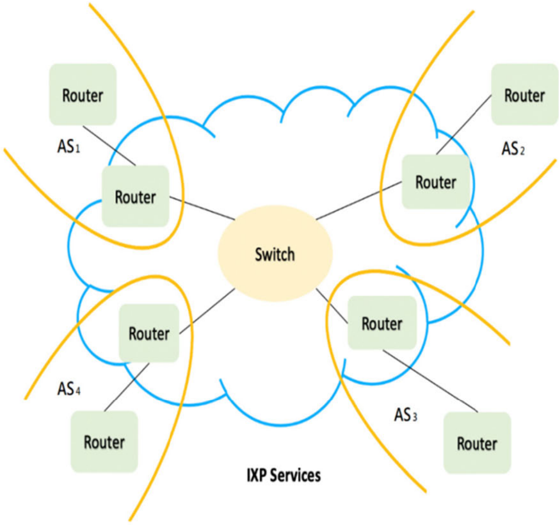
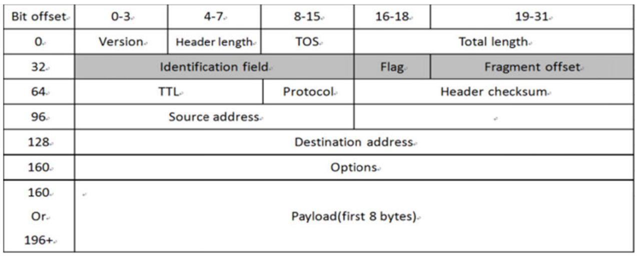


FIGURE 1. Core network with IXP service area.

interface of the router may be connected to an edge router of an autonomous system. Many IP tracing methods unable to traceback attack sources when the packets are going through the IXP framework. For example, when an attack packet is sent from an edge router of the autonomous system AS_1, it traverses the IXP switch, enters the edge router of another autonomous system AS_3, and then detours to the destination, as shown in Figure 1. When the IP tracing method traces the source of that attack packet to the edge router of AS_3, it may infer an incorrect source, such as the edge router of AS_2 or AS_4, due to more than one upstream path or changes in the table of the switch, and therefore be unable to accurately locate the attacker's device. The attack methods may also render tracebacks impossible to trace. Furthermore, attackers also attempt to evade tracing by designing a special mark or using other methods when an attack packet is sent, causing tracing methods to become ineffective. Therefore, prior to detailing the tracing method in this study, the attack model of the attacker must first be defined to determine the attack types that the proposed method can withstand:

- Multiple attackers simultaneously launching one or multiple attacks from various locations.
- Attackers simultaneously spoof their IP and MAC addresses.
- Attackers specifically fabricate a spoofed packet mark to mislead the trace direction.

We assume that attackers launch multiple exploits of the same victim from multiple locations or from one-to-many Denial-of-Service (DoS) attacks. Therefore, the method proposed in this study must be able to trace multiple attack sources simultaneously. When a router forwards a packet, it uses only the destination address to determine the downstream router that requires forwarding and does not verify the location of the source IP. Therefore, attackers can impersonate the source IP to hide their locations. We also assume that this traceback is public. Therefore, the attacker attempts to fabricate a spoofed packet mark when sending the packet and thereby render the attack source impossible to trace with network layer traceback. Although, during the transmission of the packet, the MAC address is changed to that of the router after the edge router is traversed, the spoofed MAC address nevertheless renders the link layer traceback based on MAC traceback unable to trace the attack source. Therefore, the attack source device must be inferred under the assumption that the MAC address is spoofed or modified by the attacker.



Bit offset	0-3	4-7	8-15	16-18	19-31
0	Version	Header length	TOS	Total length	
32	Identification field			Flag	Fragment offset
64	TTL		Protocol	Header checksum	
96	Source address				
128	Destination address				
160	Options				
160 Or 196+	Payload(first 8 bytes)				

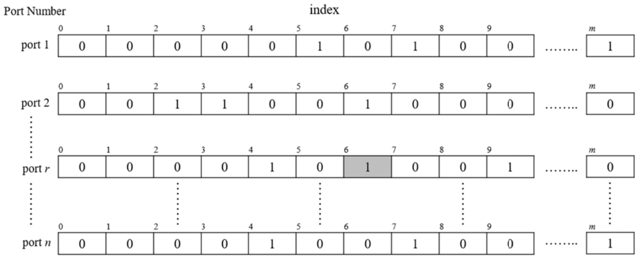
FIGURE 2. Marked fields in the packet IP header.

To trace the source locations of attackers that meet the aforementioned conditions and to define the applicability of this method, the traceback of this study must fulfil the following conditions:

- Routers and switches are secure and can resist intrusion by attackers.
- Routers know whether a packet is from the local area network (LAN) or a core network.
- The network topology does not change frequently.

The proposed algorithm marks packets as they traverse the router and performs logging as they traverse the switch. To correctly traceback the attack source, as in other studies, the routers and switches were assumed to be secure and would not be intruded to ensure that attackers would not intrude into these network devices and modify or destroy the contents of the mark and log table, which would render the attack source impossible to trace with the traceback.

The security of the routers and switches was assumed to be reasonable because, if attackers possess the ability to compromise these network devices, they possess the ability to perform more advanced attacks than we expected to prevent. Because the time used for a packet to pass through a network device, arrive at the destination end, and be detected as an attack packet is usually merely a few seconds, the researchers believed that, in most situations, the network topology would not change in such a short period of time and therefore would not cause the original network device port to differ from the upstream device. The researchers investigated methods to traceback the source of the attacker, primarily, and assumed that the victim end could detect attacks; methods to detect intrusions are not discussed in this section. However, to trace the attacker, space is required for marking. The researchers employed the IP header, as shown in the identification field, flag, and fragment offset in Figure 2, which was a total of 32 bits of space to mark path information. When a packet



Port Number	0	1	2	3	4	5	6	7	8	9		m
port 1	0	0	0	0	0	1	0	1	0	0	....	1
port 2	0	0	1	1	0	0	1	0	0	0	....	0
port *r*	0	0	0	0	1	0	1	0	0	1	....	0
port *n*	0	0	0	0	1	0	0	1	0	0	....	1

FIGURE 3. Example of *n*∗*m* two-dimensional packet log table in a switch with n ports.

enters the LAN or IXP service range, in comparison to the recording of attack information on the packet in the network layer, the link-layer packet header has no similar methods for recording path information and does not affect the space for packet transmission.

Therefore, for the packet transmission path information of the link layer, packet logging was employed to generate a digest as the index of a table using the 20 bytes of the IP packet header in the link-layer data and the 12 bytes that included the source MAC address and destination MAC address.

This digest was used to record a packet's traversing of a switch, and the packet log table also recorded the port from which the packet entered this switch. Because the researchers recorded two types of information, an $n*m$ two-dimensional packet log table was required, as shown in Figure 3, where n is equal to the number of ports of this switch and m refers to the size of the bloom filter. When a packet enters from port number r, the index value calculated by its packet header content is x, and the location of the r-th column and x-th row of the packet log table is set to 1. For example, if a packet enters from port 3 and the calculated index of that packet is 6, as shown in the gray field in Figure 3, the value of the field in the log table is set to 1 to record that packet entry from port 3.The proposed traceback comprises two main stages. The first stage is the mark/log stage, and the second is the path reconstruction stage.

A. MARKING AND LOGGING MECHANISMS

To trace the true source from which an attacker launched an attack and to solve the dilemma of IXP services in the core network, which may render the stealth IP traceback method ineffective, the researchers proposed a novel hybrid multilayer, multisource stealth forensic traceback method for attack sources that combines network-layer IP tracing and link-layer MAC tracing.

To resolve the complication whereby these devices process only the link-layer packet header, link-layer tracing technology was employed in this study. TAP was used for packet mirroring for the switch, and the packet log table, and the bloom filter was used to record the port of the switch from which the packet had entered to traceback the packet source in the link layer [44], [46] [47]. To efficiently identify the movement path of attack packets traversing the entire Internet and entering the LAN, the researchers proposed a packet traceback that integrates the network layer and link layer to trace the attacker device sending attack packets. Table 1 is the table of symbols required for the proposed method.

TABLE 1. Table of symbols.

R_i	$\{R_1, R_2, \ldots, R_i, \ldots, R_n\}$, routers on the path from source to destination.
P	Packets on the network.
AS_i	$\{AS_1, AS_2, \ldots, AS_i, \ldots, AS_n\}$, autonomous system on the path from source to destination.
ASN_i	AS_i identifier
L3P	Network layer portion of packet P
L2P	Link layer portion of packet P
UI_i	Interface identifier of router connected to the upstream
M_AS	AS-level packet mark of packet P
M_AS_{new}	Packet mark calculated by M_AS
M_AS_{pre}	Previous AS-level packet mark inferred from M_AS
M	Router-level packet mark of packet P
M_{new}	Packet mark calculated from M
M_{pre}	Previous router-level packet mark inferred from M
H()	Hash function
CLS()	Circular left shift
CLR()	Circular right shift
floor()	Bracket function
TTL	TTL field in packet

To save the marking space required to encode a path and also take into account the accuracy of the traceback source, the proposed IP tracing method combines use of router-level and AS-level IP tracebacks. Because an attacker may launch an attack from a device in the source's autonomous system (AS) to a victim device in the destination AS, router-level tracebacks must be implemented at the source AS and the destination AS to ensure that the attack-launching device can be accurately located even if the attacker or victim does not traverse the gateway router of the AS. Excluding the two AS of the source and destination, the packet begins only from the gateway router of the source, traverses the gateway routers of all the intermediate AS, and enters the gateway router of the destination. Therefore, when the packet traverses other AS between the routers, an AS-level traceback is employed

```
Router-Level Mark Scheme:
Input: $P$
begin
    if $P$ comes from local network
        $M = 0$
        if $R_i$ is ingress router && $R_i$ belongs to
    Source AS or Destination AS
            TTL = 255
            $M_{new} = [1 / (M + 2)] + UI_i + 1$
            TTL = TTL – 1
        end if
    end if
end
Forward packet to the next router
```

FIGURE 4. Router-level marking scheme.

in which marking is required only on the gateway router of the AS to save marking space required for encoding routers. When the packet traverses the IXP service and LAN of the core network location, because these network segments use switches to forward this packet, network-layer technology can no longer be used to process the packet.

The network-layer marking algorithm is composed of two types of tracebacks: AS-level packet marking and router-level packet marking. AS-level packet marking uses the hash value generated by the AS number passing through the hash function to perform XOR with the packet mark (initial value 0) contained in the received packet and then performs a circular left shift on the XOR result. The circular left shift is performed to prevent elimination of two identical packet mark values on the same path due to XOR. To save space for encoding marks, router-level packet marking is performed only on the source AS and the destination AS: After the packet leaves the source AS, other AS does not calculate router-level marks between routers until the packet reaches the destination AS.

When router R_i receives a packet P, it first determines whether the packet is from the LAN. If it is, R_i is the first router the packet has encountered. The router first initializes the packet mark; namely, it sets the router-level packet mark and AS-level packet mark as 0 and the TTL of the packet as the maximum value to avoid the attacker carefully designing a packet mark value when sending the packet. The purpose of this initial setting was to cause this nonzero packet mark to enter the network and fail to correctly determine the trace stop time and trace the incorrect attacker device, as well as the dilemma of different TTL initial values generated by different operating systems, which render it difficult to determine the accurate hop count that a packet has traversed.

When R_i receives a packet P that is not from the LAN, as shown in the algorithm in Figure 4, the router uses the router-level mark in the packet to first perform division and then add the identifier value of the port by which that packet entered, thereby obtaining a new mark. This mark is then written into the packet, and the packet is sent to the next router.

```
AS-Level Mark Scheme
Input: $P$
begin
    if $P$ is come from local network: // initialize
        $M\_AS_{old} = 0$
        if $R_i$ is ingress router:
            $M\_AS_{new} = \mathrm{CLS}(M\_AS_{old})$ xor $\mathrm{H}(ASN_i)$
        end if
    end if
  end
  Forward $P$ to the next router
End
```

FIGURE 5. AS-level marking scheme.

```
Switch Logging Scheme:
Input: $P$
begin
    $Index$ =H($L2P$ Header & $L2P$ first 20 bytes of
    payload from packet $P$)
    Get L2P source MAC address of packet $P$ and
    retrieve incoming port number $r$  from the
    MAC address table
    Bloom Filter[$r$][$Index$] = 1
End
```

FIGURE 6. Logging scheme.

If the packet P traverses the ingress router of any AS, as shown in Figure 5, that router uses the AS-level mark line in the packet to perform a circular left shift and then perform an XOR using the ASN of the current AS and the value obtained using the hash function.

When a packet enters the IXP service area, namely, when a packet encounters the switch in the core network, the switch mirrors the packet into the TAP by means of the monitoring port. Next, after the link layer of the packet is calculated, logging in the packet log table is completed. After receiving the packet, the switch first retrieves the link-layer digest in the packet and the source MAC address. The former is used for recording, whereas the latter is used to obtain the port from the MAC address table. The switch hashes the layer 2 header and the first 20 bytes of payload of the received packet to determine the index, and the location of the packet log index value of the port is set to 1, as shown in Figure 6.

Finally, if the packet arrives at a new AS and the upstream is an IXP service area, the digest value of the packet is first generated, and a packet log table is used to record the source of this packet as the IXP service area. During traceback, this is used as a criterion for determining whether to switch the link-layer traceback.

For example, Figure 7 is an example of four senders sending to the same destination via eight routers. Four AS are distributed in between, where the hash values of AS_1, AS_2, AS_3, and AS_4 are 1, 12, 21, and 23, respectively. Attacker1 reveals

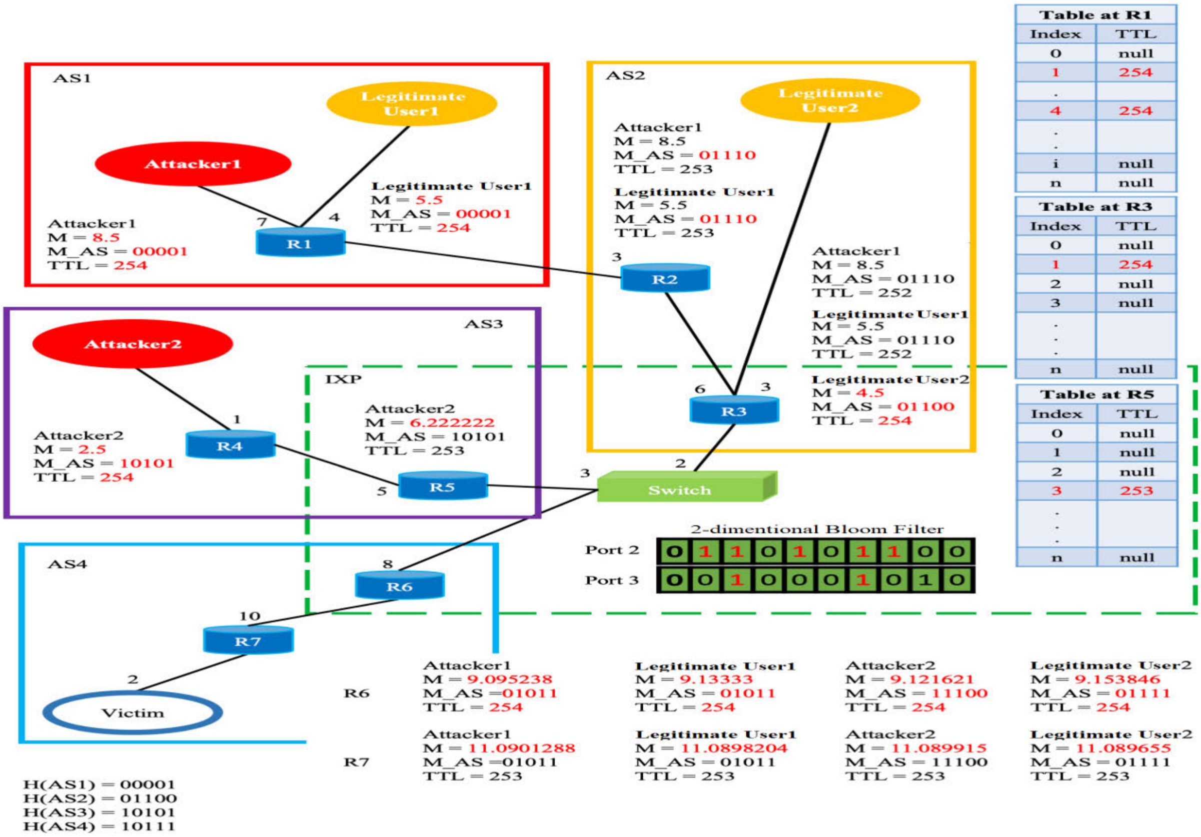


Table at R1

Index	TTL
0	null
1	254
.	
4	254
. .	
i	null
n	null

Table at R3

Index	TTL
0	null
1	254
2	null
3	null
. . .	
n	null

Table at R5

Index	TTL
0	null
1	null
2	null
3	253
. . .	
n	null

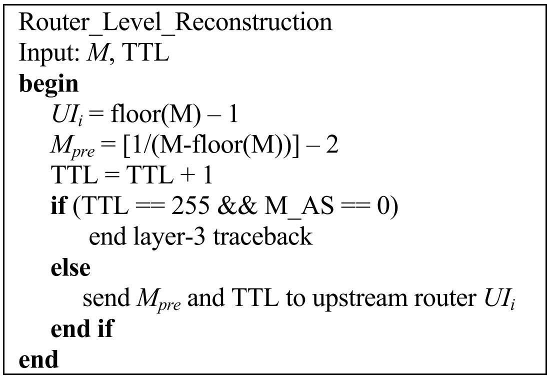
FIGURE 7. Example of two attack sources and two benign sources sent to the same destination.

that the sent packet enters the network from router R_1 and—after traversing the R_2, R_3, IXP service area, R_6, and R_7—it reaches the destination. TTL sets the value to 255 when the packet first enters R_1 and saves the current TTL when leaving the source AS to enable judgment of the distance from the attack source during path reconstruction. After the packet reaches the destination AS (AS_4), the TTL is again set to 255.

The packet sent by Attacker1 is sent to the ingress router of AS_1. Because the packet comes from the LAN, the router first initializes the router-level and AS-level marks and then uses the current AS-level packet mark to perform XOR with ASN_1 00001 passed through the hash function, yielding a value of 00001. Because AS_1 is the source AS of the packet, operations of the router-level mark are performed. When the packet is located at R_1, the packet mark and the upstream router's interface number are calculated together to obtain $[1/(M+ 2)] + UI +1 = [1/(0+2)]+7 + 1 = 8.5$, and the packet is then sent to R_2. Arriving at R_2 also means arriving at a new AS. The AS-level mark is calculated to obtain 01110, and, because the router-level mark is not located at the source AS or destination AS, the original value is maintained, and calculation is not performed. Until the packet reaches the destination AS, the TTL value in the packet is first set to 255, then the AS-level mark is calculated to yield a new value, and finally the router-level mark is calculated. Because the router does not distinguish between benign or malicious packets when marking packets, the destination also receives marked, benign packets, as shown in Figure 7. The victim receives packets sent by normal users with $M\_AS = 01011$, $M = 11.0898204$, and $M\_AS = 01111$, $M = 11.089655$, respectively. In AS_1 there is one benign user and one malicious user send packets to R_1, the router calculates the index of these two packets using the algorithm in Figure 6 and write the TTL = 4 values into the table.

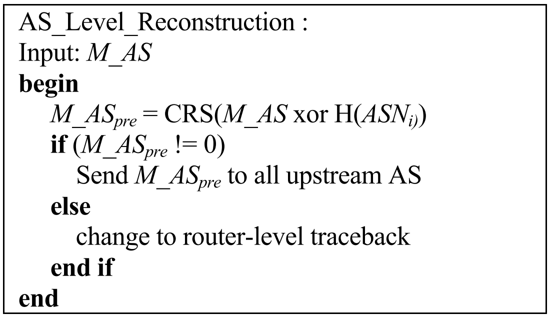
When the packet leaves the source AS, a table is first set in the egress router. This table primarily logs the TTL value of the packet. During the traceback process, when tracing reaches the source AS, TTL is extracted from this table to obtain the distance from the attack source. Before the attack packet leaves AS_1 and AS_3, as in the example Figure 7 provides, it sets a table at the respective egress routers R_1 and R_5. The packet digest values are used to pass through a hash function and calculate an index value, and the TTL value is placed into the table to which that index value corresponds.

When the packet reaches the IXP service, logging is used for the packet. The source MAC address is first used to identify the port from which the packet reached the switch (the port is 2). Next, the link-layer digest value in the packet



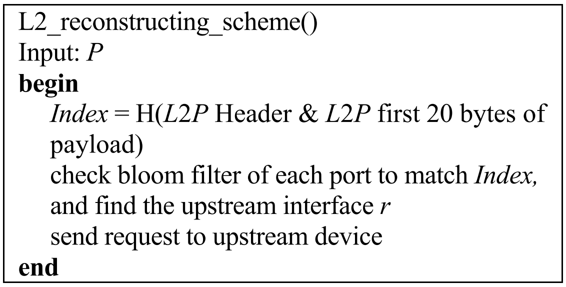
```
Router_Level_Reconstruction
Input: $M$, TTL
begin
    $UI_i$ = floor(M) – 1
    $M_{pre}$ = [1/(M-floor(M))] – 2
    TTL = TTL + 1
    if (TTL == 255 && M_AS == 0)
        end layer-3 traceback
    else
        send $M_{pre}$ and TTL to upstream router $UI_i$
    end if
end
```

FIGURE 8. Router-level reconstruction algorithm.



```
AS_Level_Reconstruction :
Input: $M\_AS$
begin
    $M\_AS_{pre}$ = CRS($M\_AS$ xor H($ASN_i$))
    if ($M\_AS_{pre}$ != 0)
        Send $M\_AS_{pre}$ to all upstream AS
    else
        change to router-level traceback
    end if
end
```

FIGURE 9. AS-level reconstruction algorithm.



```
L2_reconstructing_scheme()
Input: $P$
begin
    $Index$ = H($L2P$ Header & $L2P$ first 20 bytes of
    payload)
    check bloom filter of each port to match $Index$,
    and find the upstream interface $r$
    send request to upstream device
end
```

FIGURE 10. Layer-2 reconstruction algorithm.

is calculated, and the hash function is used on this digest value to obtain an index value (as Figure 7 depicts). The index value field in the packet log table corresponding to port 2 is then set to 1.

B. PATH RECONSTRUCTION

When a victim suffers an intrusion, the victim terminal searches for an attack source and activates a traceback. The traceback reconstructs the path according to the log table in the switch and the mark and TTL in the packet. First, the router-level traceback is shown in Figure 8. The mark is used with the floor() function to obtain the interface number of the upstream router, the UI obtained is substituted into an inverse function to calculate the previous router-level mark, and —from the UI — the traceback continues further into the previous layer. When the TTL is equal to 255 and the AS-level mark is equal to 0, this indicates that the attack path has been found, and the network-layer traceback is stopped.

The router-level algorithm performs only calculations at the source AS and the destination AS. In other environments, an AS-level traceback is employed. Its algorithm is shown in Figure 9. The researchers first use the current ASN for the hash function and then perform XOR with M_AS. Next, a circular left shift is performed to obtain the mark made by the previous AS. When the router-level mark is calculated to the destination AS edge router, it sends a traceback request to all upstream AS until an M_AS is equal to 0, revealing the correct AS traceback path.

Figure 11 shows the paths of two attackers during reconstruction. When the victim terminal detects an attack, it activates a traceback to trace the attack source. The victim terminal first performs calculations for the packet marks of the two attack packets ($M = 11.0901288$, $M\_AS = 01011$) and ($M = 11.089915$, $M\_AS = 11100$) according to the formula of the algorithm. The two router-level packet marks are passed through the floor() function, revealing that the interface numbers of the upstream routers are all 10 and that the packet marks marked by the previous router are ($M = 9.095238$) and ($M = 9.121621$). After this information is obtained, the calculated marks can be sent to the upstream device using the upstream routers' interface numbers and calculation continued for devices that are further upstream. The AS-level packet mark undergoes XOR and a circular left shift to obtain ($M\_AS = 01110$) and ($M\_AS = 10101$). The router-level mark traces back to R_6, which means it has reached the AS_4 edge router. After the router-level mark is calculated, the packet digest value is used to determine whether the upstream is an IXP service area. If it is, it switches to a link-layer traceback. If it is not, a traceback message is sent to each upstream AS.

According to the figure, because the calculation result is derived from a packet from the IXP service area, a switch is performed to a link-layer traceback to continue traceback. After entering the IXP service area, the link-layer digest value in the packet passes through the hash function for the index value to be calculated, and the packet log table corresponding to every port in the switch is searched to obtain the upstream router's interface number. A traceback at the network layer is then continued.

After the packet of Attacker1 leaves the IXP service area, it first arrives at an AS. After an AS-level mark is calculated to obtain the value 01110, a request is sent to all AS upstream of the AS until one of the AS calculates an AS-level mark value of 0; this indicates that the traceback has reached the source AS. The subsequent traceback continues as a router-level traceback.

III. ANALYSIS OF STORAGE CAPACITY AND ACCURACY

In this section, the accuracy of the algorithm is discussed. First, the experimental environment of this method is introduced.

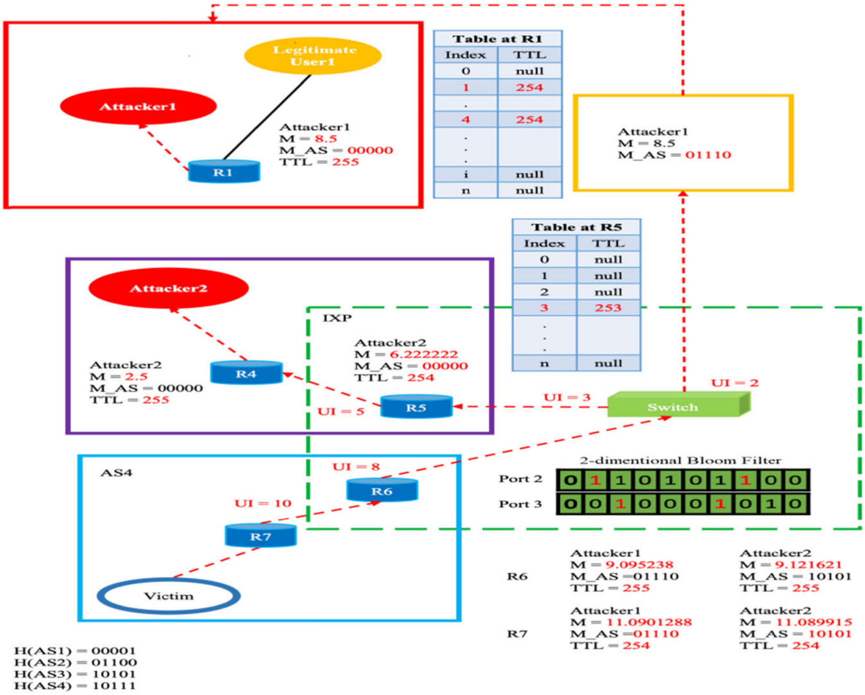


Table at R1	
Index	TTL
0	null
1	254
.	
4	254
. . .	
i	null
n	null

Table at R5	
Index	TTL
0	null
1	null
2	null
3	253
. . .	
n	null

2-dimentional Bloom Filter

Port 2	0	1	1	0	1	0	1	1	0	0
Port 3	0	0	1	0	0	0	1	0	1	0

FIGURE 11. Reconstruction of attack path.

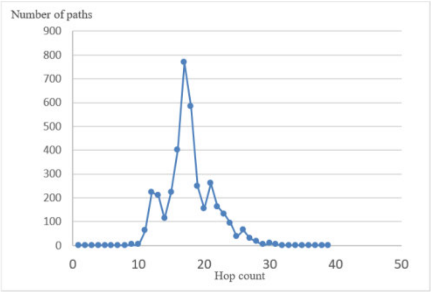


FIGURE 12. Path length distribution.

A. EXPERIMENTAL ENVIRONMENT

To analyze the storage requirements for the router, CAIDA's Ark ITDK was used to generate a network topology. The Ark data set was composed of multiple IP paths generated by traceroute. Because some routers may not respond to pings, some path data were incomplete. Therefore, only 3,804 complete paths were taken from the data to establish the network topology required. The analysis results of Ark data path length are shown in Figure 12. These data had a total of 10,222 routers, 661 AS, an average path length of 17.74, and 16 IXP in the topology.

B. ANALYSIS OF STORAGE CAPACITY

For this method, a log table is set at the edge router of the source AS to store the digest value and TTL value of a packet. Its primary purpose is to enable the traceback to use the TTL value when tracing the source, allowing it to more accurately find the true attack source. The space expended by logging increases as the number of packets increases. Immediately after a packet had exited IXP services, the index value represented by the digest value of that packet was also recorded, and space required for logging was evaluated according to the actual network topology.

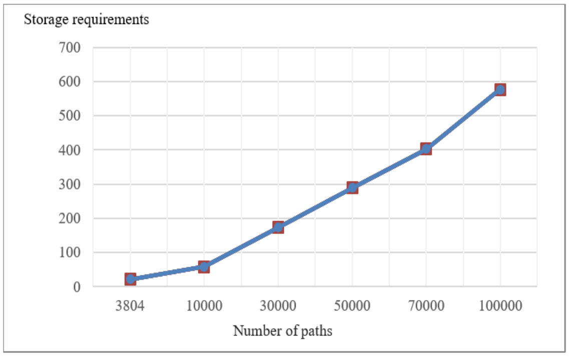


FIGURE 13. Analysis of router storage capacity.

In our system, a n*m two-dimensional bloom filter was used for logging, where n is equal to the number of ports of this switch and m refers to the size of the bloom filter. Figure 13 is a diagram of the relationship for log table size and the number of paths. In the experiment, a path was randomly taken and 3804, 10000, 30000, 50000, 70000, and 100000 were repeated. Since the packets from same path carry the same digest, the storage is path specific and not based on the number of packets passing through it. Figure 13 indicates that more paths required more logging space. The storage space required for 100,000 paths was

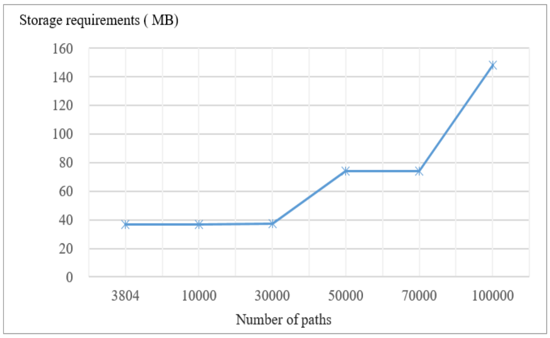


FIGURE 14. Analysis of router and switch total storage capacity.

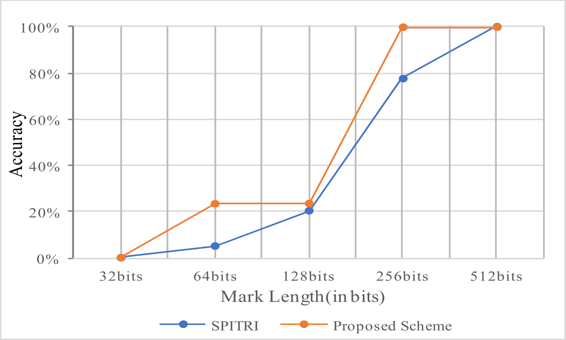


FIGURE 15. Comparison of accuracy between Vijayalakshmi's method and method proposed in the present study in a network environment without IXP services.

approximately 576 KB. In addition, in an environment containing a switch, a 2-dimensional bloom filter was used for logging. The size selected for each bloom filter was twice the number of paths.

Figure 14 is a diagram of the relationship between the size of the logging space required by the router and the space consumed by a 2-dimensional bloom filter and its path tree in a switch environment. Because the design of the 2-dimensional bloom filter required much of logging space, the logging space required at 100,000 paths was approximately 148 MB.

C. ANALYSIS OF ACCURACY

This section compares the researchers' method with the SPITRI technique of Vijayalakshmi *et al.* [48] in two types of network environments: one that does not include IXP services and one that does. To analyze the accuracies of the two methods in attack source traceback in conditions of different encoding space sizes, the mark lengths in each method were also modified to different lengths for comparison. Figure 15 shows the accuracy of various mark lengths in a network environment without IXP services for the method in the present study and for the SPITRI method of Vijayalakshmi [48]. When the mark length was 32 bits, the accuracy of the two methods was relatively low primarily due to the precision error of the floating-point number. After the mark length was uniformly increased to 256 bits, accuracy increased significantly. Because the method of the present study combines the AS-level and router-level, at the same path length, fewer division operations cause errors than for the SPITRI method proposed by Vijayalakshmi. Therefore, the attack source can be identified more efficiently. When the encoding space was less than 256 bits, compared to the path length in the CAIDA data, the method proposed in the present study exhibited limited advantages due to insufficient space. However, when the space exceeded 256 bits, even if the core network contains IXP services, the method proposed in the present study continued to exhibit higher accuracy.

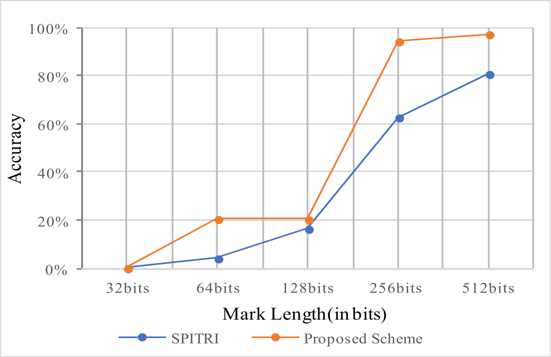


FIGURE 16. Comparison of accuracy of Vijayalakshmi's method and method proposed in the present study in a network environment with IXP services.

When a network environment contains IXP services, Vijayalakshmi's SPITRI method results in a relationship between routers that is not one-to-one. This is because of switches in the network environment—meaning that it is unable to accurately identify the true attack source. Figure 16 shows the accuracy of various mark lengths for Vijayalakshmi's SPITRI method [48] and the method proposed in the present study in a network environment containing IXP services. In the IXP services, we use the two-dimensional bloom filters to log the packet, and it does not affect on the accuracy of the IP marking scheme. The amplitude of the accuracy curve reveals that the method proposed in the present study is not substantially changed by the inclusion of IXP services in a network environment. Nevertheless, when mark length was at 256 bits, due to the sufficiently large encoding space, the accuracy of the method proposed in the present study exhibited a distinct advantage over that of Vijayalakshmi's method.

IV. CONCLUSION

This article proposes a single-packet traceback combining network-layer and data link-layer tracebacks. In the switch, TAP mirroring packets are employed for logging, and TTL is used to determine the termination of the traceback. The number of routers from the attack source can thus be obtained. Even if a core network includes a switch, a single packet can be used to accurately traceback the attack source.

Although some space is sacrificed to log packet information, the traceback is no longer unable to find the true attack source due to a core network containing switches. The method proposed in this study is compared to that proposed by Vijayalakshmi. At the same mark length, the method proposed in this study can reduce the number of operations and decrease the error probability caused by division operations. In a network environment with IXP services, the method proposed in this study nevertheless correctly identifies the attack source.

However, the method proposed in this study cannot prevent disadvantages such as excessive information stored in the log table, which causes resource exhaustion or collision. Because the IP packet header's mark length was only 32 bits, to substantially increase traceback accuracy, the mark could be cut into pieces to be placed in different packets.

REFERENCES

[1] B. Krebs. *Hacked Cameras, DVRs Powered Today's Massive Internet Outage*. Accessed: Jul. 6, 2020. [Online]. Available: https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage

[2] P. Paganini. *Car Hacking—Chinese hacker team remotely hacked Tesla Model S*. Accessed: Jul. 6, 2020. [Online]. Available: http://securityaffairs.co/wordpress/51469/hacking/tesla-model-s-hack.html

[3] N. Bindra and M. Sood, ''Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset,'' *Autom. Control Comput. Sci.*, vol. 53, no. 5, pp. 419–428, Nov. 2019, doi: 10.3103/S0146411619050043.

[4] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, ''IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?'' *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, Sep. 2018, doi: 10.1109/MSP.2018.2825478.

[5] L. Xiao, D. Xu, N. B. Mandayam, and H. V. Poor, ''Attacker-centric view of a detection game against advanced persistent threats,'' *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2512–2523, Nov. 2018, doi: 10.1109/TMC.2018.2814052.

[6] L. F. Sikos, ''Packet analysis for network forensics: A comprehensive survey,'' *Forensic Sci. Int., Digit. Invest.*, vol. 32, Mar. 2020, Art. no. 200892, doi: 10.1016/j.fsidi.2019.200892.

[7] A. Belenky and N. Ansari, ''Accommodating fragmentation in deterministic packet marking for IP traceback,'' in *Proc. IEEE Global Telecommun. Conf.*, 2003, pp. 1374–1378, doi: 10.1109/GLOCOM.2003.1258463.

[8] A. Belenky and N. Ansari, ''IP traceback with deterministic packet marking,'' *IEEE Commun. Lett.*, vol. 7, no. 4, pp. 162–164, Apr. 2003, doi: 10.1109/LCOMM.2003.811200.

[9] A. Belenky and N. Ansari, ''Tracing multiple attackers with deterministic packet marking (DPM),'' in *Proc. IEEE Pacific Rim Conf. Commun. Comput. Signal Process.*, 2003, pp. 49–52, doi: 10.1109/PACRIM.2003.1235716.

[10] V. K. S. Rajam and S. Mercy Shalinie, ''A novel traceback algorithm for DDoS attack with marking scheme for online system,'' in *Proc. Int. Conf. Recent Trends Inf. Technol.*, Chennai, Tamil Nadu, 2012, pp. 407–412, doi: 10.1109/ICRTIT.2012.6206751.

[11] H. Tian, J. Bi, and P. Xiao, ''A flow-based traceback scheme on an AS-level overlay network,'' in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2012, pp. 559–564, doi: 10.1109/ICDCSW.2012.49.

[12] V. Aghaei-Foroushani and A. N. Zincir-Heywood, ''IP traceback through (authenticated) deterministic flow marking: An empirical evaluation,'' *EURASIP J. Inf. Secur.*, vol. 2013, no. 1, Dec. 2013, doi: 10.1186/1687-417X-2013-5.

[13] H. Takurou, K. Matsuura, and H. Imai, ''IP traceback by packet marking method with Bloom filters,'' in *Proc. 41st Annu. Int. Carnahan Conf. Secur. Technol.*, Oct. 2007, pp. 255–263, doi: 10.1109/CCST.2007.4373498.

[14] A. Yaar, A. Perrig, and D. Song, ''FIT: Fast Internet traceback,'' in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies.*, 2006, pp. 1395–1406, doi: 10.1109/INFCOM.2005.1498364.

[15] D. Xiaodong Song and A. Perrig, ''Advanced and authenticated marking schemes for IP traceback,'' in *Proc. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2001, pp. 878–886, doi: 10.1109/INFCOM.2001.916279.

[16] H. Tian, J. Bi, X. Jiang, and W. Zhang, ''A probabilistic marking scheme for fast traceback,'' in *Proc. 2nd Int. Conf. Evolving Internet*, Sep. 2010, pp. 137–141, doi: 10.1109/INTERNET.2010.32.

[17] J. Liu, Z.-J. Lee, and Y.-C. Chung, ''Dynamic probabilistic packet marking for efficient IP traceback,'' *Comput. Netw.*, vol. 51, no. 3, pp. 866–882, Feb. 2007.

[18] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, ''Network support for IP traceback,'' in *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 226–237, Jun. 2001, doi: 10.1109/90.929847.

[19] V. Paruchuri, A. Durresi, and S. Chellappan, ''TTL based packet marking for IP traceback,'' in *Proc. IEEE Global Telecommun. Conf.*, 2008, pp. 1–5, doi: 10.1109/GLOCOM.2008.ECP.490.

[20] S. Saurabh and A. S. Sairam, ''Linear and remainder packet marking for fast IP traceback,'' in *Proc. 4th Int. Conf. Commun. Syst. Netw.*, Jan. 2012, pp. 1–8, doi: 10.1109/COMSNETS.2012.6151318.

[21] Y. Bhavani, V. Janaki, and R. Sridevi, ''IP traceback through modified probabilistic packet marking algorithm,'' in *Proc. IEEE Int. Conf. IEEE Region*, Oct. 2013, p. 10, doi: 10.1109/tencon.2013.6718523.

[22] Z. Zhou, B. Qian, X. Tian, and D. Xie, ''Fast traceback against large-scale DDoS attack in high-speed Internet,'' in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Dec. 2009, pp. 1–7, doi: 10.1109/CISE.2009.5363316.

[23] X. Liu, M. Dong, K. Ota, L. T. Yang, and A. Liu, ''Trace malicious source to guarantee cyber security for mass monitor critical infrastructure,'' *J. Comput. Syst. Sci.*, vol. 98, pp. 1–26, Dec. 2018, doi: 10.1016/j.jcss.2016.09.008.

[24] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, ''Single-packet IP traceback,'' *IEEE/ACM Trans. Netw.*, vol. 10, no. 6, pp. 721–734, Dec. 2002, doi: 10.1109/TNET.2002.804827.

[25] L. Zhang and Y. Guan, ''TOPO: A topology-aware single packet attack traceback scheme,'' in *Proc. Securecomm Workshops*, Baltimore, MD, USA, 2006, pp. 1–10, doi: 10.1109/SECCOMW.2006.359556.

[26] T. Korkmaz, C. Gong, K. Sarac, and G. Sandra Dykes, ''Single packet IP traceback in AS-level partial deployment scenario,'' *Int. J. Secur. Netw.*, vol. 2., no. 1, pp. 95–108, Mar. 2007, doi: 10.1504/IJSN.2007.012828.

[27] E. Hilgenstieler, E. P. Duarte, G. Mansfield-Keeni, and N. Shiratori, ''Extensions to the source path isolation engine for precise and efficient log-based IP traceback,'' *Comput. Secur.*, vol. 29, no. 4, pp. 383–392, Jun. 2010, doi: 10.1016/j.cose.2009.12.011.

[28] C. Gong and K. Sarac, ''A more practical approach for single-packet IP traceback using packet logging and marking,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 10, pp. 1310–1324, Oct. 2008, doi: 10.1109/TPDS.2007.70817.

[29] K. H. Choi and H. K. Dai, ''A marking scheme using Huffman codes for IP traceback,'' in *Proc. 7th Int. Symp. Parallel Architectures, Algorithms Netw.*, Hong Kong, 2004, pp. 421–428, doi: 10.1109/ISPAN.2004.1300516.

[30] S. Malliga and A. Tamilarasi, ''A hybrid scheme using packet marking and logging for IP traceback,'' *Int. J. Internet Protocol Technol.*, vol. 5, nos. 1–2, pp. 81–91, Apr. vol. 2010.doi, p. 10.1504/IJIPT.2010.032617.

[31] S. Malliga and A. Tamilarasi, ''A proposal for new marking scheme with its performance evaluation for IP traceback,'' *WSEAS Trans. Comput. Res.*, vol. 3, no. 4, pp. 259–272, Apr. 2008.

[32] Y. Wang, S. Su, Y. Yang, and J. Ren, ''A more efficient hybrid approach for single-packet IP traceback,'' in *Proc. 20th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process.*, Garching, China, 2012, pp. 275–282, doi: 10.1109/PDP.2012.38.

[33] M.-H. Yang and M.-C. Yang, ''RIHT: A novel hybrid IP traceback scheme,'' *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 789–797, Apr. 2012, doi: 10.1109/TIFS.2011.2169960.

[34] N. Lu, Y. Wang, F. Yang, and M. Xu, ''A novel approach for single-packet IP traceback based on routing path,'' in *Proc. 20th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process.*, Feb. 2012, pp. 253–260, doi: 10.1109/PDP.2012.40.

[35] M. H. Yang, ''Hybrid single-packet IP traceback with low storage and high accuracy,'' *Sci. World J.*, vol. 2014, pp. 1–12, 2014, doi: 10.1155/2014/239280.

[36] V. Praveena, S. Karthik, and G. Jeon, ''Hybrid approach for IP traceback analysis in wireless networks,'' *Wireless Pers. Commun.*, vol. 113, no. 1, pp. 669–690, Jan. 2020, doi: 10.1007/s11277-020-07183-8.

[37] M. H. Yang, M. C. Yang, J. Luo, and W. C. Hsu, ''High accuracy and low storage hybrid IP traceback,'' in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jeju, China, 2014, pp. 1–5, doi: 10.1109/CITS.2014.6878977.

[38] A. Saini, C. Ramakrishna, and S. Kumar, "A hybrid optimization algorithm based on ant colony and particle swarm algorithm to address IP traceback problem," in *Cognitive Informatics and Soft Computing*. Singapore: Springer, 2019, pp. 429–439, doi: 10.1007/978-981-13-0617-4_43.

[39] Z. Ling, J. Luo, D. Xu, M. Yang, and X. Fu, "Novel and practical SDN-based traceback technique for malicious traffic over anonymous networks," in *Proc. IEEE Conf. Comput. Commun.*, Paris, France, 2019, pp. 1180–1188, doi: 10.1109/INFOCOM.2019.8737586.

[40] C. Li, F. Hu, and D. Xu, "RPDT: An architecture for IP traceback in partial deployment scenario," in *Proc. IEEE 5th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2019, pp. 1602–1608, doi: 10.1109/ICCC47050.2019.9064353.

[41] T. Baba and S. Matsuda, "Tracing network attacks to their sources," *IEEE Internet Comput.*, vol. 6, no. 2, pp. 20–26, 2002, doi: 10.1109/4236.991439.

[42] H. Hazeyama, O. Masafumi, and Y. Kadobayashi, "A Layer-2 extension to hash-based IP traceback," *IEICE Trans. Inf. Syst.*, vols. E86–D, no. 11, pp. 2325–2333, Nov. 2003.

[43] M. Snow and J. Park, "Link-layer traceback in Ethernet networks," in *Proc. 15th IEEE Workshop Local Metrop. Area Netw.*, Princeton, NJ, USA, 2007, pp. 182–187, doi: 10.1109/LANMAN.2007.4295996.

[44] S. Marios Andreou and A. V. Moorsel, "Logging based IP traceback in switched Ethernets," in *Proc. 1st Eur. Workshop Syst. Secur.*, Glasgow, Scotland, 2008, pp. 1–7, doi: 10.1145/1355284.1355286.

[45] Arjmandpanah?Kalat, "Design and performance analysis of an efficient single flow IP traceback technique in the AS level," *Int. J. Commun. Syst.*, vol. 33, no. 9, 2020, doi: 10.1002/dac.4382.

[46] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970, doi: 10.1145/362686.362692.

[47] J.-N. Luo and M.-H. Yang, "Improved single packet traceback scheme with Bloom filters," *Proc. Int. Conf. Internet Things Service*. Cham, Switzerland: Springer, 2017, pp. 177–184, doi: 10.1007/978-3-030-00410-1_21.

[48] M. Vijayalakshmi and S. Mercy Shalinie, "Single packet ICMP traceback technique using router interface," *J. Inf. Sci. Eng.*, vol. 30, no. 6, pp. 1673–1694, 2014. [Online]. Available: https://jise.iis.sinica.edu.tw/JISESearch/pages/View/PaperView.jsf?keyId=9_116, doi: 10.1688/JISE.2014.30.6.1.

MING-HOUR YANG (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University. He is currently a Professor of information and computer engineering with Chung Yuan Christian University. He is also the Board Supervisor of Chinese Cryptology and Information Security Association (CCISA) and the Editorial Board Member of *International Journal of Information Systems and Social Change*. He was the Secretary General of CCISA. He served as the Guest Editor of *International Journal of Security and Networks and Information Security Newsletter*. He also served as a Program Committee Member of DEXA'16, AsiaJCIS'16, OBD'16.

JIA-NING LUO (Member, IEEE) received the Ph.D. degree in computer science of National Chiao Tung University, Taiwan. He is currently an Associate Professor and the Chairman with the Department of Information and Telecommunications Engineering, Ming Chuan University, Taiwan. He serves as the Director for the Chinese Cryptology and Information Security Association (CCISA). He also served as a Program Committee Member for the IEEE DSC'18, AsiaJCIS'20 and the IEEE DSC'21. His research interests include network security, authentication protocols, the IoT security and eWallet security.

M. VIJAYALAKSHMI received the Ph.D. degree in information and communication engineering from Anna University, India. She is currently an Associate Professor with the Department of Computer Science and Engineering, Thiagarajar College of Engineering, India. She serves as a Reviewer to several journals including the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, *IET Information Security*, *Security* (Wiley), and *Communication Networks*. She also served as a Program Committee Member of SICBS 2017 and the IEEE DSC 2018. Her research interests include network security, digital forensics, and the Internet of Things.

S. MERCY SHALINIE (Senior Member, IEEE) is currently a Professor and the Dean of the Thiagarajar College of Engineering, Madurai, India. Her research interests include AI, machine learning and information security. She has the distinction of publishing over 100 research articles in refereed International and National Journals and Conferences. Her sustained research interest has made her complete sponsored Research and Development projects from DRDO, AICTE, DeitY, DST, NTRO, Honeywell, and Yahoo, India. Her passion to work in Free/Open Source Software has led to the development of ICT Framework for Thiagarajar College of Engineering which has received National level accolades. She has Postdoctoral Research experience with the University of California, Irvine, USA, and Monash University, Melbourne, Australia. She is a Senior Member of CSI, IE, ACCS, and ISTE.