Hash Based AS Traceback against DoS Attack

Sharmin Aktar
Department of Computer Science
University of New Orleans
New Orleans, LA, USA 70148
Email: saktar@uno.edu

Abdullah Yasin Nur
Department of Computer Science
University of New Orleans
New Orleans, LA, USA 70148
Email: ayn@cs.uno.edu

***Abstract*—The design of IP protocol creates difficulties in identifying the true source of any packet, which makes it one of the most arduous problems to defend against Denial of Service (DoS) attacks. This paper introduces an Autonomous System (AS) traceback mechanism based on probabilistic packet marking, which allows the victim to trace the attack-originating AS. Traceback on the AS level has several advantages containing a reduced number of routers involvement for packet marking as well as the required number of packets to infer the forward path. We utilize the IP packet header to implement our packet marking methodology. Our results show that a victim site can trace the attack path with 33.25 packets on average. Additionally, we provide an encoding method to significantly reduce the false cases in path reconstruction.**

***Index Terms*—Denial of Service attack, DoS, DDoS, AS Traceback, IP Traceback**

I. Introduction

Despite being a beneficial source of communication among users, the immense growth and usage of the Internet have endangered our cyber world by making it vulnerable to attack. Denial of Service (DoS) attack has recently raised a major security concern in cyberspace. In this attack, the adversary aims to make the victim's network unavailable to legitimate users by interrupting its regular services. Usually, attackers perform DoS attacks by sending excessive requests towards the targeted machine in order to overload its system. In a Distributed Denial of Service (DDoS) attack, which is more devastating than the DoS attack, the victim is being bombarded by several compromised machines or botnets that make the attack more difficult to defend. Most recently, Amazon Web Services (AWS) encountered an attack with a peak volume of 2.3 Tbps in February 2020 [6].

The defense mechanism against DoS attacks has been classified into four phases; intrusion prevention, detection, response, and finally tolerance and mitigation [3]. The first stage of the method targets to stop the attack from being initiated. Some of the well-known strategies of this phase include ingress/egress filtering and change of target IP address. In the second phase, the system tries to detect the attack by comparing it with some known attack signatures. However, the victim site may fail to recognize the attack when the database used in detection has not been properly updated. After the intrusion detection, the quick response is to locate the source of the attack. The last phase focuses on reducing the impact of the attack to improve the quality of service. Among the four phases, the third one is more promising as it tries to detect the attack source. Once the real source is identified, further control can be imposed more specifically. However, attackers frequently use IP Spoofing techniques to bypass their original identity, making this step more complex. Hence the traceback problem appears where we need to find out the source of the attack origin.

IP traceback is a challenging problem because of the IP protocol's stateless nature and having no source address validation check on IP packets. It is defined as determining the IP address path taken by any packet transferred from the attacker site towards the victim on Internet [4]. Packet marking is a technique where routers blemish their IP address into the IP packet's header, and the destination site collects blemished packets to discover the forward path from the source. Another version of the IP traceback is Autonomous System (AS) traceback, where the AS path is being traced between an attacker and the victim site instead of the IP address path. An Autonomous System is a large network or a collection of networks controlled by a single organization. Generally, an Internet Service Provider (ISP) or a large administrative entity like governmental agency operates the AS [5]. Every AS on the Internet is assigned a unique number called Autonomous System Number (ASN). ASN used to be an unsigned 16-bit integer. The drastic increase in the number of ASes in the Internet required a change in ASNs, where 32-bit numbers were introduced in 2007.

In this paper, we propose a probabilistic AS level traceback method by utilizing the IP header field of the IP packet. Instead of tracing all IP addresses between an attacker and a victim site, our approach detects only the ASes between them. AS traceback has several advantages over IP traceback, including reducing path reconstruction time and the number of required packets to rebuild the attack path. In our approach, we introduce a similar marking approach in [2] by using the hashing-based edge encoding method. Instead of inserting the IP address of routers, we encode their ASN hashes. As several routers belong to an AS, marking the same AS by every router is not necessary. Therefore, only border routers belonging to different ASes are used for marking purposes. We have utilized a flag bit to save the marking status of any packet in the same path to avoid overwriting. In case a previous marker is

978-1-6654-0306-1/21/$31.00

found to be marked in a packet, the current router executes an XOR operation between the hashes of both ASN and updates the edge information. A distance field is used to preserve the order between the marking router and the victim. After the victim receives all distanced edge information, it proceeds to reconstruction by doing repetitive XOR operations until it finds the attacker ASN. We induce an encoded version of our method where we used the ASN type as the deciding factor of using hashing. Upon using the encoding, we reduced the false-negative cases significantly.

We conducted several experiments using a real world dataset to demonstrate the efficiency of our approach under DoS attacks. Our results show that a victim site can construct the AS level forward path from an attacker site after receiving 33.25 packets on the average. Particularly, the victim can construct the path with 1 packet for AS hop distance 1 and 58.87 packets on the average for AS path 11. Additionally, we compared our results to other Probabilistic Packet Marking (PPM) scheme for AS Traceback [14]. Our results show that PPM needs 23.02 to 96.08 packets on the average to construct AS level forward paths for varying hop distances. Our method requires 1 to 50.87 packets on the average for the same AS paths. PPM can track a maximum of 8 AS hop distance, whereas our method can track up to 16 AS hop distance. Our maximum packet count is 69.05 for 11 AS hop length, which is still less than PPM's required packet numbers for 8 hop distance. On the other hand, the PPM method uses 16-bit, and we use 21 bits in the IP packet header. Therefore, our approach consumes slightly more bandwidth.

The rest of the paper is organized as follows. In Section II, we discuss the related works. A detailed explanation of our methodology is presented in Section III. Section IV demonstrates our experimental results. Finally, we conclude the paper in Section V.

II. Related Work

Researchers have proposed several defense mechanisms against DoS attacks over two decades [10, 20]. One of the suggested solutions named IP traceback is about finding the paths between the attackers and the victim site. Existing methods for traceback can be categorized as reactive and proactive types [1]. Reactive tracebacks are initialized once an attack has been identified, and the process is workable until the attack is alive. On the contrary, proactive methods capture tracing information while packets are routed via the network. The captured data is then used for reconstructing the attack path once enough information has been traced. Packet marking falls into the proactive category, which has been used in most traceback approaches.

One of the earliest packet marking schemes proposed by Savage et al. [7] had drawn widespread attention. They introduced a compressed edge fragment method by using 16-bit IP identification field to mark the fragmented router's data in a probabilistic manner. After receiving sufficient packets, the victim site reconstructs attack paths by using marked packets. By using their approach, most paths could be resolved with between one and two thousand packets.

Song and Perrig proposed Advanced and Authenticated Marking Schemes (AMS), which is a modified marking approach inspired by Savage's work [2]. AMS decreases both the number of required packets and the false-negative rates for constructing the forward paths. They also use the 16-bit IP ID field divided into three parts: distance, edge, and hash function fields. Every router's IP address is being converted into a set of independent hash values while marking. If a router chooses to mark a packet, it marks the packet header with the hash value of its IP address. In case the packet is already marked by a previous router, the router executes an XOR operation between the current edge on the packet and the hash of its own IP address. It overwrites the XOR result in the edge field in the packet header.

Nur and Tozal proposed a novel probabilistic packet marking scheme to derive forward paths from attackers to a victim site by using the Record Route feature of the IP protocol [4]. In their work, each router checks whether there is enough space for a new entry in the options field. If the options field of the packet is not full, the router appends its IP address. On the other hand, if the options field is full, a router probabilistically restarts record routing by clearing the options field of the IP packet header. The victim site gradually constructs the attack path by incorporating the sub-paths from the received packets.

Murugesan et al. [8] introduced a single-packet IP traceback scheme called "HPSIPT" which comprises both logging and packet marking. It introduces two marking fields that require 40 bits. Each router's incoming and outgoing interface ID is encoded to mark value, whereas it uses a hash table of hash tables for logging purposes. In the traceback system, the reverse encoding method has been applied to retrieve the outgoing interface ID of the upstream router. The traceback process is continued until the outgoing interface ID connects to the attacker's LAN.

Bhavani et al. [9] designed an IP traceback system via a modified probabilistic packet marking algorithm using the Record-Route option in IP protocol. In their method, each router either probabilistically marks a packet by encoding its IP address with distance equals one or by doing XOR operation with the upstream router's IP address along with incrementing distance in the option field. Once a packet reaches the victim, marking data is being saved in a table called RT. Upon checking on distance value, upstream routers' IP addresses are being recovered and the attack path is revealed.

Cheng et al. [11] proposed a store-and-mark based traceback system called opportunistic piggyback marking (OPM). It is formed on message fragmentation where they use a local buffer to temporarily store fragmented messages along with their respective destinations. Once all received traceback message fragments reach the target, it groups the message fragments with the same identifiers. Thus, the end-host retrieves all the traceback messages.

Patel and Jinwala presented an authenticated packet marking approach where the router chooses marking probability p such

a way so that packet nearer to the source will have more probability [12]. In case an attacker forges the marking, the nearest router will overwrite it. If a router decides to mark a packet, both the IP address named as label and its hash value are stored in the packet header. Otherwise, it updates the marking with the XOR operation between its IP address and the label of the marking. For the reconstruction of the attack path, IP addresses are derived upon verification on the marked IP addresses with their hashes.

The popularity of the IP traceback generated another approach which is called AS traceback. In this process, only Autonomous Systems on the attack path are being inferred. The number of ASes between two end hosts is below the number of routers because several backbone routers may involve delivering the packets within the same AS. Therefore, AS traceback techniques require fewer packets to detect the path between a source and the destination.

Paruchuri et al. [13] designed an AS level traceback mechanism where 16-bit ASN has been added into the packet's IP Identification field by the ASBR based on some fixed probability. However, the change in ASN from 16-bit to 32-bit made this method obsolete. Gao and Ansari had implemented an AS-based Edge Marking (ASEM) [18] method based on an optimal probabilistic model. Their marking technique uses the BGP routing table information where only the ingress edge routers of each AS took part in marking. Okada et al. [14] proposed a 32-bit ASN based traceback system with a fixed probability where they applied hashing on the fragments of ASN for encoding purpose. Alenezi and Reed [21] proposed a similar approach with a dynamic probability to reduce the number of required packets. They utilize 25 bits in the IP header and use BGP tables to obtain AS hop distance between the destination and the current router. The rewrite probability is calculated dynamically based on the AS Path distance from BGP. Nur and Tozal [17] use the Record Route option field of IP protocol to discover the path with a single packet. Their tradeoff is using more bandwidth to reduce the number of required packets and remove false cases.

III. Methodology

In our proposed method, we revisited and improved the legacy work from Savage's probabilistic packet marking [2]. Instead of tracing the IP path between two end hosts, we trace Autonomous Systems by tracking Autonomous System Numbers (ASN). Autonomous System (AS) in the Internet is defined as a group of networks administered by one or more network operators under a well-defined routing policy. Each AS is assigned a 32-bit unique identifier which is called ASN.

As stated in [17], there are 15.43 Interface level hops between two end hosts in the Internet, whereas those IP interfaces belong to 4.16 ASes on the average. Therefore, instead of tracing all the router's IP Addresses between an attacker and a victim, we encode their ASN in the packet header. Tracing AS Path instead of Interface path has many benefits, such as reducing the number of required packets for attack path reconstruction and minimizing the router's marking overhead. Unlike the IP traceback schemes, only the Autonomous System Border Router (ASBR) takes part in our marking method and the other core routers skip marking packets. More specifically, the first router that belongs to a different AS on a path participates in the marking process.

In our marking scheme, we store the edge information between two adjacent Autonomous Systems on the attack path. We utilize 21 bits in an IP packet header, including the 16-bit IP Identification and 5-bit Type-of-Service fields. Note that IP ID and ToS fields are rarely used in the current Internet [15, 16]. We utilize a 16-bit hash function to store the ASN edges. Additionally, we use 1-bit to indicate a marked packet within a trace and 4 bits for keeping the AS distance between the victim and the router being marked. Previous work [17] shows that the AS level diameter of the Internet is 12. Therefore, we use 4 bits for the distance field, which can track up to 16 ASes between two end hosts.

The detailed marking process of our method is explained in the Algorithm 1. Similar to [2], we assume that the victim site has the knowledge of the upstream map of ASes. When a packet is being forwarded to an Autonomous System Border Router (ASBR), it first checks two conditions shown in line 3. One is the probability p, and another one is the $flag$ bit, which we use for prohibiting any subsequent marks by further routers if that packet is already being marked on the same trace. In case an ASBR decides to mark the packet, it calculates the 16-bit hash of its ASN, $h(ASN)$, and writes the hash value to the $edge$ field of the packet with setting 0 to the $distance$ field, which is stated in line 4-6. Line 7-10 demonstrates the latter case, which is the marked case. The $distance$ 0 states that the packet is already marked by a previous router. The current ASBR calculates the XOR value of its ASN's hash value and the $edge$ field value stored on that packet. It overwrites the edge field with the result of the XOR in line 9. ASBRs increase the $distance$ field regardless of the marking decision so that the victim site can find the distance of the ASBR that marked the packet. The $edge$ field, which stores the XOR value of two subsequent ASBR, encodes an edge between the two neighboring ASes in the upstream AS map. By backpropagating the XOR result starting with the victim AS, the previous ASes can be decoded.

For illustration purposes, Figure 1 shows an example attack path. Dashed lines indicate internal links and hide intermediate backbone routers since they do not involve in the marking process. ASBRs are marked as *marking router* in the figure, which are responsible for marking. The marking scenario is displayed in the figure where different ASes contribute to marking in each trace. For example, when a packet traverses through the *marking ASBR* of $AS1103$, it will mark the packet based on probability p by storing $h(1103)$ in the edge field and $distance$ with 0. The next marking router in $AS11357$ executes $XOR(h(1103), h(11357))$ since the packet is marked by the previous ASBR. Any subsequent *ASBR* will ignore marking except increasing the distance field. Therefore, the victim receives a packet containing the edge field with value $XOR(h(1103), h(11357))$ and $distance$

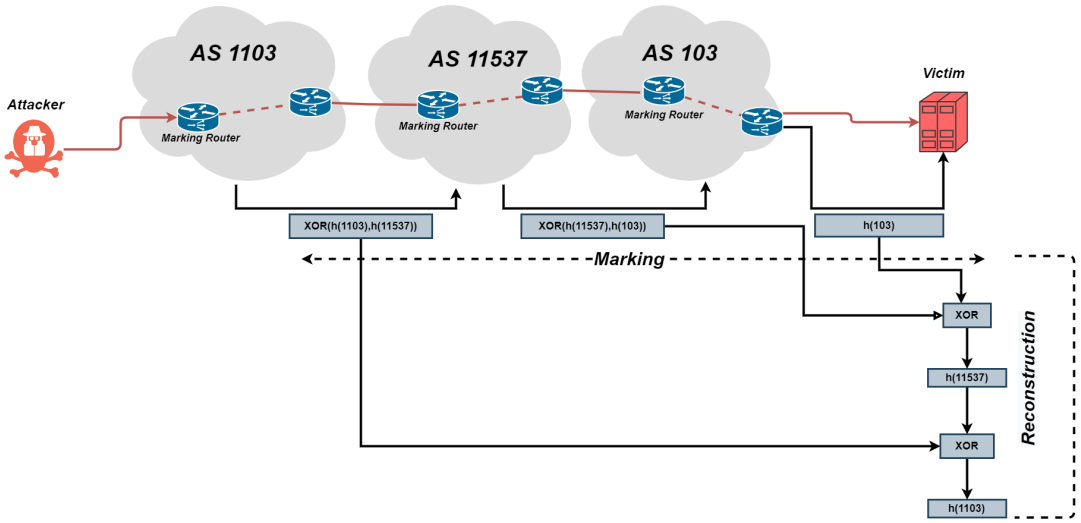

Fig. 1: An Example of Marking and Reconstruction Procedure for Our Method

Algorithm 1 Marking Process

1: for each packet P
2: let r be a random number from $[0, 1)$
3: **if** $r <$p and $P.flag = 0$ **then** ▷ p is the probability for marking
4: $P.distance = 0$
5: $P.edge = hash(ASN)$ ▷ produces a 16-bit hash of ASN
6: $P.flag = 1$ ▷ $flag = 1$ indicates a marked packet
7: **else** ▷ router chooses not to mark the packet
8: **if** $P.distance = 0$ **then**
9: $P.edge = XOR(P.edge, hash(ASN))$ ▷ edge encoding part
10: $P.distance = P.distance + 1$

Algorithm 2 Reconstruction of AS Path

1: let $path$ be empty for $[0, d-1]$▷ d=no.of AS between attacker & victim
2: $path[0] = VictimASN$
3: **for** $i \leftarrow 0$ to $d-1$ **do**
4: $x = XOR(EdgeSet[i+1], hash(path[i]))$ ▷ EdgeSet stores the distance wise edge-information
5: **while** $path[i]$ has child y **do**
6: **if** $hash(y) = x$ **then**
7: $path[i+1] = y$
8: $break$

of 2. In another packet, the victim receives the $edge$ as $XOR(h(11357), h(103))$ and $distance$ of 1 when the *marking ASBR* from 2nd AS will take part in marking. Finally, the victim receives the marking from the last AS'es marking router with $h(103)$ and $distance$ of 0. When all possible distanced edge values are available to the victim, the victim site reconstructs the forwarding path.

Algorithm 2 describes the reconstruction procedure of our approach. Similar to [2], we assume that the victim has knowledge of the upstream AS map of the Internet. Thus, the victim uses the upstream map as a graph, G rooted at the victim and starts the reconstruction method from the root. The $path$ field denotes the set of ASes at a distance d from the victim in the reconstructed attack graph. After getting the necessary markings, the victim site will have the all distanced edge list, $EdgeSet$. The victim site inserts its ASN in the $path$ at the distance 0 in line 2. For each ASN in the path, the victim computes XOR between upstream $edge$ and hash of the ASN at this distance, $XOR(EdgeSet[i+1], hash(path[i]))$ noted in line 4. The XOR will result in the hash value of the next distanced ASN on the $path$, stated as $path[i+1]$. The victim then checks whether the hash of any child AS, y of $path[i]$ in G matches with the computed XOR. If the victim finds a matched ASN, then it adds it to the path. These steps are being repeated until it reaches the attacker's site. Thus, the victim reconstructs the attack graph.

An example of the reconstruction process has been shown in Figure 1. In the attack case, the victim is 3 AS-hop away from the attacker, so the victim can start the reconstruction process once it gets all the edges of distance $[0, 2]$. In the first step, the victim site inserts $ASN103$ to $path[0]$ with a distance of 0. Then, it will start the process from its AS'es hash value, $h(103)$ and will execute the XOR operation as described in the Algorithm 2. The output of $XOR(XOR(h(11357), h(103)), h(103))$ equals to $h(11357)$. The victim site verifies the calculated hash value by checking upon each AS connected to its AS. When it finds a correct match, 11357 will be put in the $path[1]$. After that, next path value will be retrieved by doing the $XOR(XOR(h(1103), h(11357)), h(11357))$ and verifying in the same manner. Thus, victim will find the reconstructed attack path as: $\{d_0 = 103, d_1 = 11357, d_2 = 1103\}$.

A. Encoding ASNs

Until 2007, Autonomous System Numbers were 16-bit unsigned integer. The drastic increase in the number of Autonomous Systems in the Internet required a change in ASNs. Internet Assigned Numbers Authority (IANA) replaced the 16-bit ASN with 32-bit ASNs in 2007. Many early ASes keep their 16-bit ASN by padding 0 to the beginning to make it 32-bit.

Algorithm 3 Marking Process With Encoding

```
1: function HAHSINGWITHENCODING(ASN)
2:     if ASN.length <= 16 then
3:         return ASN
4:     else
5:         return hash(ASN)
6: for each packet P
7: let r be a random number from [0, 1)
8: h ⟵ HAHSINGWITHENCODING(ASN)
9: if r <p and flag = 0 then          ▷ p is the probability for marking
10:    P.distance = 0
11:    P.edge = h
12:    P.flag = 1                     ▷ flag=1 indicates a marked packet
13: else                              ▷ router chooses not to mark packet
14:    if P.distance = 0 then
15:        P.edge = XOR(P.edge, h)
16:    P.distance = P.distance + 1
```

Algorithm 4 Reconstruction With Encoding

```
1: function HAHSINGWITHENCODING(ASN)
2:     if ASN.length <= 16 then
3:         return ASN
4:     else
5:         return hash(ASN)
6: let path be empty for [0, d − 1]▷ d=no.of AS between attacker & victim
7: path[0] = VictimASN      ▷ assume victim will know the ASN itself
8: for i ← 0 to d − 1 do
9:     h ⟵ HAHSINGWITHENCODING(path[i])
10:    x = XOR(EdgeSet[i + 1], h)  ▷ EdgeSet stores the distance wise
   edge-information
11:    while path[i] has child y do
12:        h′ ⟵ HAHSINGWITHENCODING(y)
13:        if h′ = x then
14:            path[i + 1] = y
15:            break
```

In our method, we use 16-bit field to keep hash values of ASN. Therefore, it produces false cases in the reconstructed paths. Based on our experiments, we observe that out of 682,301 unique path traces, 37,831 false cases have been found in the basic version of our method. In order to reduce the false cases and improve the performance, we developed an encoding technique. In case of an ASN can be represented by 16-bits, we discard the hashing part and store ASN directly. By applying this part in our methods, we reduce the false cases from 37,831 to 8,312 path traces.

Algorithm 3 and Algorithm 4 refer to the corresponding marking and reconstruction process of the encoded version. Line 1-5 describes a function *"HASHINGWITHENCODING"* where we have differentiated between the ASN type. If the ASN can be represented by 16-bit, we use the ASN value directly. Otherwise, we use the hash value of the ASN.

B. Probabilistic Model

Assume that d is the AS level hop distance between an attacker to the victim site. Similar to [7], the expected number of packets required for the victim to reconstruct a path is bounded by Equation 1.

$$E[X] < \frac{ln(d)}{p \times (1-p)^{d-1}} \tag{1}$$

Our goal is to minimize the number of required packets to reconstruct the attack path. Therefore, we need to minimize the upper bound value in equation 1. To minimize the function, we need to maximize the value of $p \times (1-p)^{d-1}$. Since the maximization problem can be solved by applying Fermat's interior extremum theorem, we take the first derivative of the function and set it to zero. Equation 2 shows that the optimal value of p is inverse of the AS distance between an attacker and the victim site. We used the probability p as $\frac{1}{11}$ since the maximum AS hop distance between two end hosts is 11 in our dataset.

$$\frac{\partial}{\partial(p)}[(p \times (1-p)^{d-1})] = 0$$

$$(1-p)^{d-1} - (d-1) \times p \times (1-p)^{d-2} = 0$$

$$(1-p)^{d-1} = (d-1) \times p \times (1-p)^{d-2}$$

$$\frac{(1-p)^{d-1}}{(1-p)^{d-2}} = (d-1) \times p$$

$$1 - p = p \times (d-1)$$

$$p = \frac{1}{d} \tag{2}$$

IV. EXPERIMENTAL RESULTS

In this section, we empirically demonstrate the efficiency of our algorithm using a real-world dataset. We used the CAIDA IPv4 Prefix-Probing Traceroute Dataset [22] consisting of more than 69 million path traces. Additionally, we used Route-Views prefix to AS mapping dataset obtained from CAIDA [23]. In order to generate an AS Level Internet topology, we mapped IP addresses reported in the traceroute dataset to their corresponding ASes. The dataset consists of 682,301 AS level path traces with 42,548 different ASes. The minimum and maximum AS level hop lengths in our dataset are 1 and 11, respectively. The average AS level hop length is 4.28. The hop length distribution of our dataset has been shown in Figure 2.

In our experimental setup, we assumed that each AS path trace is an attack path where the source of the trace is the attacker and the destination is the victim site. We generated a graph from the AS traceroute dataset where the root is the victim site. The generated graph is used as the upstream AS map for the victim to verify the hash stored in the marking field. We have executed our methods in two approaches, the basic version without encoding and the modified version with

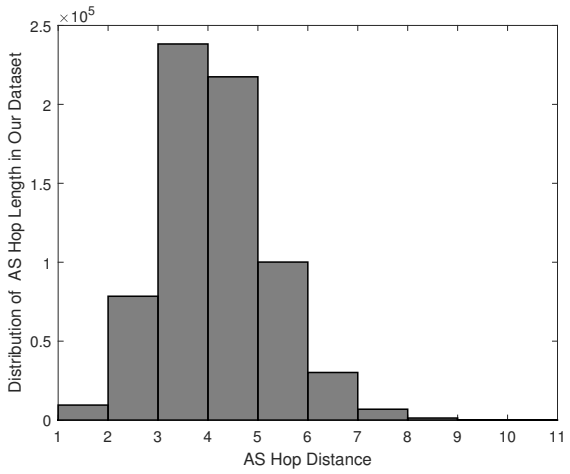

Fig. 2: AS Level Hop Length Distribution

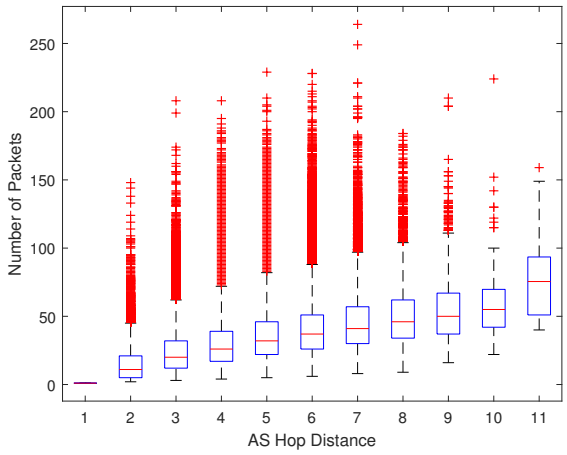

Fig. 4: Actual Number of Required Packets per Hop Distance for Attack Path Reconstruction

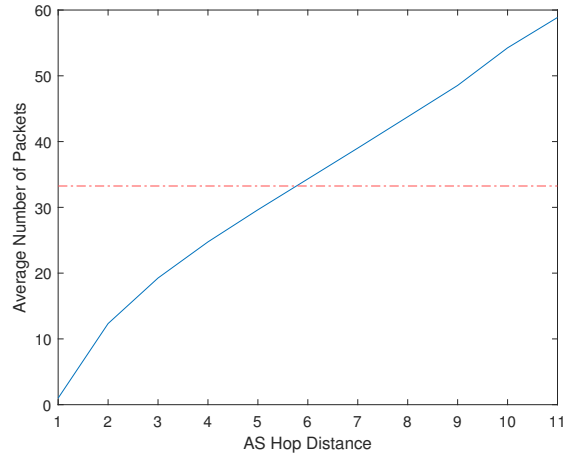

Fig. 3: Average Number of Packets for Reconstruction

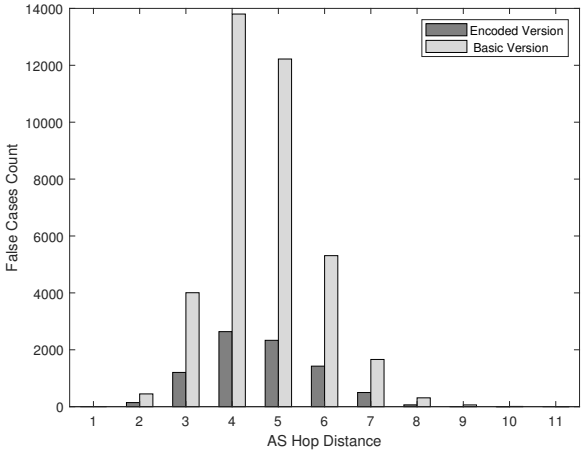

Fig. 5: False-Negative Comparison

encoding. In the basic approach, we used hashing regardless of ASN length, while in another version, we utilized the length factors. The details of our approaches have been described in the Section III.

Experimental results show that our method with encoding can trace the routes with 33.25 packets on the average. Figure 3 shows the average number of packets needed to construct a forward path with respect to AS level hop distances between the attacker and victim sites. The red line in the figure shows the overall average at 33.25 packets. Particularly, it reaches to 58.87 at AS hop distance 11. To provide the reader with more insight, Figure 4 shows the box plot of required number of packets per hop distance.

In the second part of the experiment, we investigate the false-negative cases. Remember that we introduce an encoding technique to reduce false cases. Figure 5 presents a comparison between our basic version without encoding and the modified version with encoding. We observe that out of 682,301 cases, 37,831 AS path traces have at least one false case in the basic version. On the other hand, we observe only 8,312 false cases in the modified version, corresponding to 1.21% of our dataset.

A. Comparison with Other AS Traceback Methods

In the following, we show the efficiency of our method by comparing it with another Probabilistic Packet Marking (PPM) based AS traceback method. We implemented the method in [14] using a fixed probability of 0.092, as suggested in the paper. Note that the distance field in the PPM method is represented by 3 bits, giving the maximum number of traceable AS counts as 8. On the other hand, we use 4 bits for distance, giving us the ability to track a maximum of 16 AS hop distance.

Figure 6 shows that PPM method in [14] needs 23.02 to 96.08 packets on the average to construct AS level forward paths for varying hop distances. Our method requires 1 to 43.76 packets on the average for the same paths. Additionally, the maximum AS hop distance in our dataset is 11, where the PPM method is not able to discover, but our method can

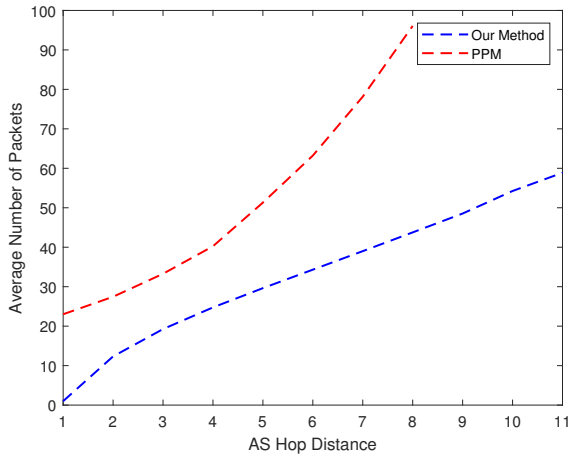

Fig. 6: Comparison with PPM based AS traceback method

discover the path with 58.87 packets on the average. On the other hand, the PPM method uses 16-bit, and we use 21 bits in the IP packet header. Therefore, our approach consumes slightly more bandwidth.

V. Conclusions

Denial of Service (DoS) attack poses an alarming threat to our increasing cyberspace. Tracing the packet's source is an effective way of defending against DoS attacks, as most attackers use IP spoofing to hide their identities. In this work, we proposed a hash-based AS traceback system to deduce the attack paths from attacker sites to a victim site. Traceback on the AS level has several advantages containing a reduced number of routers involvement for packet marking as well as the number of packets. We exploited the IP header to implement our packet marking methodology. In the proposed technique, only border routers of different ASes take part in the marking process, which reduces the router involvement. Experimental results show that our method can trace the attack path with 33.25 packets on average. Furthermore, we proposed an encoding technique that reduces false cases remarkably.

References

[1] H. Aljifri, "IP Traceback: a New Denial-of-Service Deterrent?" IEEE Security & Privacy, 2003

[2] D. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," IEEE INFOCOM 2001

[3] C. Douligeris and A. Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art", Computer Networks, Vol. 44, Issue 5, 2004

[4] A. Y. Nur and M. E. Tozal, "Record Route IP Traceback: Combating DoS Attacks and the Variants", Computers & Security, Volume 72, Pages 13-25, 2018

[5] B. Donnet and T. Friedman, "Internet Topology Discovery: a Survey," IEEE Communications Surveys & Tutorials, vol. 9, no. 4, pp. 56-69, 2007

[6] "Amazon 'thwarts largest ever DDoS cyber-attack'", BBC News. Jun 18, 2020. Retrieved Nov 11, 2020

[7] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback", IEEE/ACM Transactions on Networking 9.3 (2001): 226-237

[8] V. Murugesan, M. Selvaraj, and M. Yang, "HPSIPT: A High-Precision Single-Packet IP Traceback Scheme", Computer Networks, vol. 143, no. 2018, pp. 275-288, 2018

[9] Y. Bhavani, V. Janaki, and R. Sridevi, "IP Traceback Through Modified Probabilistic Packet Marking Algorithm Using Record Route", International Conference on Computational Intelligence and Informatics (ICCII), 2018

[10] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems", ACM Computing Surveys, 2007

[11] L. Cheng, D. M. Divakaran, W. Y. Lim, and V. L. L. Thing, "Opportunistic Piggyback Marking for IP Traceback," IEEE Transactions on Information Forensics and Security, vol. 11, no. 2, pp. 273-288, Feb. 2016

[12] H. Patel and D. Jinwala, "LPM: A Lightweight Authenticated Packet Marking Approach for IP Traceback", Computer Networks, pp. 41-50, 2018

[13] V. Paruchuri, A. Durresi, R. Kannan, and S. Iyengar, "Authenticated Autonomous System Traceback", IEEE International Conference on Advanced Information Networking and Application, 2004

[14] M. Okada, Y. Katsuno, A. Kanaoka, and E. Okamoto, "32-bit AS Number Based IP Traceback", IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011

[15] I. Stoica and H. Zhang, "Providing Guaranteed Services without per Flow Management", ACM SIGCOMM Computer Communication Review, 29.4, pp. 81-94, 1999

[16] A. Durresi, V. Paruchuri, and L. Barolli, "FAST: Fast Autonomous System Traceback", Journal of Network and Computer Applications, pp. 448-454, 2009

[17] A. Y. Nur and M. E. Tozal, "Single Packet AS Traceback against DoS Attacks", IEEE International Systems Conference (SysCon), 2021

[18] Z. Gao and N. Ansari, "A Practical and Robust Inter-domain Marking Scheme for IP Traceback", Computer Networks, 51.3, pp. 732-750, 2007

[19] M. Alenezi and M. J. Reed, "Selective Record Route DoS Traceback," International Conference on Risks and Security of Internet and Systems (CRiSIS), pp. 1-7, 2013

[20] A. Y. Nur, "Combating DDoS Attacks with Fair Rate Throttling", IEEE International Systems Conference (SysCon), 2021

[21] M. Alenezi and M. J. Reed, "Traceback of DoS over Autonomous Systems", International Journal of Network Security and Its Applications, 5.2, 2013

[22] CAIDA Prefix-Probing Traceroute Dataset - 2021/01/01, http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml

[23] Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6 - http://www.caida.org/data/routing/routeviews-prefix2as.xml