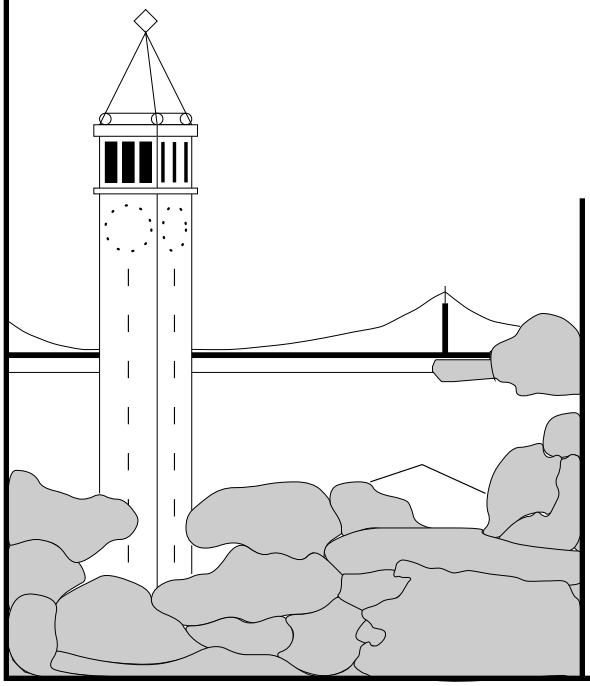
Advanced and Authenticated Marking Schemes for IP Traceback

Dawn X. Song and Adrian Perrig



Report No. UCB/CSD-00-1107

June 2000

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Cloudflare, Inc. Exhibit 1014 Page 1

Advanced and Authenticated Marking Schemes for IP Traceback*

Dawn Song and Adrian Perrig
dawnsong,perrig@cs.berkeley.edu
Computer Science Department
University of California, Berkeley

Abstract

Denial-of-service attacks are among the hardest security problem on today's Internet. One difficulty to thwart these denial-of-service attacks is to trace the source of the attacks because they often use incorrect, or "spoofed" IP addresses to disguise the true origin. In this paper, we present two new schemes, the Advanced Marking Scheme and the Authenticated Marking Scheme, which allow the victim to traceback the approximate origin of spoofed IP packets. Our techniques have very low network and router overhead and support incremental deployment. Comparing with previous work, our techniques have significantly higher precision (lower false positive rate) and lower computation overhead for the victim to reconstruct the attack paths under large scale distributed denial-of-service attacks. Furthermore the Authenticated Marking Scheme provides efficient authentication of routers' markings such that even a compromised router cannot forge or tamper markings from other uncompromised routers.

1 Introduction

Denial-of-service (DoS) attacks pose an increasing threat to today's Internet [How98]. Even more concerning, automatic attacking tools such as Tribal Flood Network (TFN) and TFN2K allow teenagers to launch widely distributed denial-of-service (DDoS) attacks with just a few keystrokes [Tea00]. Just to name one of the many cases, in February 2000, several high-profile sites including Yahoo, eBay, and Amazon were brought down for hours by DDoS attacks [Tea00]. A serious problem to thwart these DoS attacks is that attackers use incorrect, or "spoofed" IP addresses in the attack packets and hence disguise the real origin of the attacks. Due to the stateless nature of the Internet, it is a difficult problem to determine the source of these spoofed IP packets, which is called the *IP traceback* problem.

While many IP traceback techniques have been proposed, they all have significant weaknesses that limit their usability in practice (we discuss more details on related work in section 5). One promising solution, recently proposed by Savage et al. [SWKA00], is to let routers probabilistically mark packets with partial path information during packet forwarding. The victim can then reconstruct the complete paths after receiving a modest number of packets. We refer to this type of approach as the *IP marking approach*. This approach has a low overhead for routers and the network and supports incremental deployment. Unfortunately, as we will show in our theoretical analysis (appendix A) and simulation results (section 3.3), this approach has a severe computation overhead for the victim to reconstruct the attack paths and gives a large number of false positives when the denial-of-service attacks come from multiple distributed attackers. For example, this approach can require days of computation to reconstruct the attack paths and give thousands of false positives even when there are only 25 distributed attackers. This approach is also vulnerable to compromised routers. If a router is compromised, it can forge markings from other uncompromised routers and hence lead the reconstruction to wrong results. Even worse, the victim will not be able to tell a router is compromised just from the information in the packets it receives.

In this paper we present two new IP marking techniques, Advanced Marking Scheme and Authenticated Marking Scheme, to solve the IP traceback problem. Our approach has the same efficiency for routers and the network as the scheme proposed by Savage et al. [SWKA00], yet our approach is much more efficient and accurate for paths reconstruction under DDoS. In particular, our approach takes less than 100 seconds to reconstruct the attack paths and give almost no false positives even for DDoS from 1500 distributed attackers. Furthermore, our Authenticated Marking Scheme supports efficient authentication of routers' markings. This prevents a compromised router from forging markings from other uncompromised routers. Our schemes also support incremental deployment and allow the victim to reconstruct the

*We gratefully acknowledge support for this research from several US government agencies. This research was suported in part by the Defense Advanced Research Projects Agency under DARPA contract N6601-99-28913 (under supervision of the Space and Naval Warfare Systems Center San Diego), by the National Science foundation under grant FD99-79852, and by the United States Postal Service under grant USPS 1025 90-98-C-3513. Views and conclusions contained in this document are those of the authors and do not necessarily represent the official opinion or policies, either expressed or implied of the US government or any of its agencies, DARPA, NSF, USPS.

Cloudflare, Inc. Exhibit 1014 Page 2

attack paths even after the attack has completed.

The rest of the paper is organized as follows. In section 2, we review background information and highlight the main challenges of the IP marking approach. In section 3, we describe our new Advanced Marking Schemes and show theoretical analysis and experiment results which indicate our Advanced Marking Schemes are efficient and accurate even in the presence of large scale DDoS attacks. In section 4, we describe our new Authenticated Marking Scheme that provides efficient authentication of routers' markings. Finally we discuss some practical issues and the related work in section 5, and conclude in section 6.

2 Background and the Challenge

2.1 Definitions

The directed acyclic graph (DAG) rooted with V in figure 1 represents the network as seen from a victim V and a distributed denial-of-service attack from A_2 and A_3. V could be either a single host under attack or a network border device such as a firewall representing many such hosts. Internal nodes $\{R_i\}_i$ of the graph represent the routers, which we refer to as *upstream* routers from V. We refer to this graph as the *map of upstream routers from* V. For every router R_i, we refer to the set of routers that immediately before R_i in the graph as the *children* of R_i, e.g. R_3, R_6 and R_4 are R_2's children. The leaves $\{A_i\}$ represent the potential *attack origins*, or *attackers*. The *attack path* from A_i is the ordered list of routers between A_i and V that the attack packet has traversed, e.g. the two dotted lines in the graph indicate two attack paths: (R_6, R_3, R_2, R_1) and (R_7, R_4, R_2, R_1). The *distance* of R_i from V on a path is the number of routers between R_i and V on the path, e.g. the distance of R_6 to V in the path (R_6, R_3, R_2, R_1) is 3. The *attack graph* is the graph composed of the attack paths, e.g., the attack graph in the example will be the graph containing the two attack paths (R_6, R_3, R_2, R_1) and (R_7, R_4, R_2, R_1). And we refer to the packets used in DDoS attacks as *attack packets*. We call a router *false positive* if it is in the reconstructed attack graph but not in the true attack graph. Similarly we call a router *false negative* if it is in the true attack graph but not in the reconstructed attack graph. We call a solution to the IP traceback problem robust if it has very low false negatives and false positives.

2.2 IP Marking with Edge Sampling

The basic idea of the IP marking approach is that routers probabilistically write some encoding of partial path information into the packets during forwarding. A basic technique, the *edge sampling algorithm*, is to write *edge* information into the packets [SWKA00]. This scheme reserves two static fields of the size of IP address, *start* and *end*, and a static *distance* field in each packet. Each router updates these fields as the following:

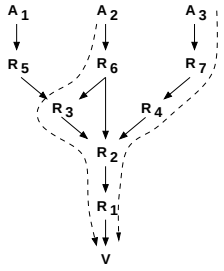


Figure 1. Upstream router map from victim

Each router marks the packet with a probability q. When the router decides to mark the packet, it writes its own IP address into the start field and writes zero into the distance field. Otherwise, if the distance field is already zero which indicates its previous router marked the packet, it writes its own IP address into the end field, thus represents the edge between its previous routers and itself. Finally, if the router doesn't mark the packet, then it always increments the distance field. Thus the distance field in the packet indicates the number of routers the packet has traversed from the router which marked the packet to the victim. The distance field should be updated using a saturating addition, meaning that the distance field is not allowed to wrap. The mandatory increment of the distance field is used to avoid spoofing by an attacker. Using such a scheme, any packet written by the attacker will have distance field great than or equal to the length of the real attack path.

The victim can use the edges marked in the attack packets to reconstruct the attack graph. For each attack path with distance d, the expectation of the number of packets needed to reconstruct the path is bounded by $\frac{ln(d)}{q(1-q)^{d-1}}$.

2.3 Overloading the IP Identification Field

The edge sampling algorithm requires 64 bits for the start and end field and another few bits for the distance field in every IP packet. We could store these bits in an IP option, but this is impractical because appending additional data to a packet on the fly is expensive and may lead to fragmentation. We could also send it in a separate packet, but this adds more router, network and compatibility overhead. A more efficient solution is to overload the 16-bit IP Identification field used for fragmentation in the IP header. Recent measurements suggest that only less than 0.25% of packets are fragmented [SZ99]. We refer to [SWKA00] for discussions on practical issues about overloading the IP Identification field.

2.4 Limitation of the Fragment Marking Scheme and The Challenge

In order to just use the 16-bit IP Identification field to store the IP markings, we need an encoding scheme to reduce the storage requirements in each packet.

Cloudflare, Inc. Exhibit 1014 Page 3

The Fragment Marking Scheme [SWKA00] proposed an encoding scheme where they split each router's IP address with some redundancy check into eight fragments and probabilistically mark the IP packet with one of the eight fragments. This encoding scheme works well with just a single attacker. But in case of a distributed denial-of-service attack, this scheme severely suffer from two main problems:

- High computation overhead, because it needs to check a large number of combinations of the fragments,
- Large number of false positives, because the redundancy check is insufficient and the false positives at a closer distance to the victim can cause even more false positives further away from the victim.

For example, as shown in our simulation results (section 3.3), even in case of a DDoS from 25 distributed attacker sites, the Fragment Marking Scheme takes days to reconstruct the attack graph and results in thousands of false positives. We also include a more detailed theoretical analysis in appendix A and simulation results in section 3.3.

The Fragment Marking Scheme also suffers from the fact that it is not robust against a compromised router. Even worse, a victim cannot even tell that a router has been compromised merely from the information in the packets received.

The real challenge is how we can design efficient, accurate and authenticated encoding schemes for IP marking using just the 16-bit space in each packet.

3 Advanced Marking Schemes

In this section, we describe our Advanced Marking schemes, in which we use new encoding schemes that are efficient and accurate even under DDoS from over 1500 distributed attacker sites. We have an observation that if the victim knows the map of its upstream routers, then it does not need full IP addresses in the marking packets to reconstruct the attacking graph, and hence the marking scheme can be more communication and computation efficient. For our marking schemes, we assume the victim has a map of its upstream routers, denoted as $\mathcal{G}_m$. $\mathcal{G}_m$ is a DAG with the victim as the root. Later in section 5.1, we show that this assumption is valid and practical.

3.1 Advanced Marking Scheme I

In the basic approach, we use a similar marking scheme as the Fragment Marking scheme, except that instead of encoding the IP address of a router R_i into eight fragments as in the Fragment Marking scheme, we simply encode it as its hash value, $h(R_i)$, as indicated in Figure 2. In this scheme, we divide the 16-bit IP Identification field into a 5-bit *distance* field and a 11-bit *edge* field, as indicated in Figure 2. Note that 5 bits can represent 32 hops which is sufficient for almost all Internet paths [CC97, TR00, CAI].

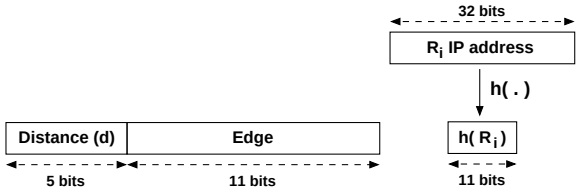


Figure 2. Encoding in Advanced Marking Scheme I

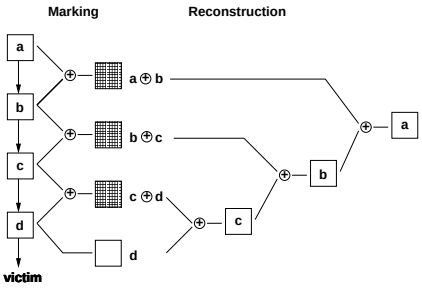


Figure 3. XOR Encoding

Marking. Figure 4 describes the marking procedure of Advanced Marking Scheme I. Note that we actually use two independent hash functions, h and h', in the encoding of the routers' IP addresses. h and h' both have 11-bit outputs. Every router marks a packet with a probability q when forwarding the packet. If a router R_i decides to mark the packet P, it writes $h(R_i)$ into the edge field and 0 into the distance field in packet P. Otherwise, if the distance field is 0 which implies its previous router has marked the packet, it XORs $h'(R_i)$ with the edge field value and overwrites the edge field with the result of the XOR. The router always increments the distance field if it decides not to mark the packet. The XOR of two neighboring routers indicate the edge between the two routers. The edge field of the marking will contain the XOR result of two neighboring routers, except for samples from routers one hop away from the victim. Because $a \oplus b \oplus a = b$, we could start from markings from the routers one hop away from the victim, and then hop-by-hop, decode the previous routers, as shown in figure 3. The reason to use two independent hash functions is to distinguish the order of the two routers in the XOR result.[1]

Reconstruction. The reconstruction procedure is described in figure 4. Intuitively, to reconstruct the attack paths, the victim uses the upstream router map $\mathcal{G}_m$ as a road-map and performs a breadth-first search from the root. Let's denote the set of edge fields marked with a distance d as Ψ_d (do not include duplicates). At distance 0, the victim enumerates all the routers one hop away from itself in $\mathcal{G}_m$ and check which routers have the hash value of their IP addresses, $h(R_i)$, matched with the edge fields in Ψ_0, and denotes the set of

[1] Given a collision-resistant hash function g, we can simply implement the two independent hash functions in a standard way: $h(x) = g(\langle 0, x\rangle), h'(x) = g(\langle 1, x\rangle)$, where $\langle _, _ \rangle$ means concatenation.

Cloudflare, Inc. Exhibit 1014 Page 4

matched IP addresses as S_0. Therefore S_0 is the set of routers one hop away from the victim in the reconstructed attack graph. S_d denotes the set of routers at distance d to the victim in the reconstructed attack graph. Then for each edge x in Ψ_{d+1}, and for each element y in S_d, the victim computes $z = x \oplus h'(y)$. The victim then checks whether any child R_i of y in $\mathcal{G}_m$ has the hash value of its IP address, $h(R_i)$, equal to x. If the victim finds a matched IP address R_u, then it adds R_u to the set S_{d+1} (initially S_{d+1} is empty). The victim repeats the steps until it reaches the maximal distance marked in the packets, denoted as $maxd$. Thus, the victim reconstructs the attack graph.

Analysis. Assume in a DDoS attack, let $|M_d|$ denotes the number of routers in the attack graph at distance d from the victim. Let t_y denote the in-degree of element y in S_{d-1} (the number of y's children) in $\mathcal{G}_m$, and recall that $|\Psi_d|$ is the number of unique edge segments received by the victim with the distance field marked as d. Because the hash value is 11 bits, the expected number of false positives among y's children in $\mathcal{G}_m$ is $t_y \cdot |\Psi_d|/2^{11}$. If assume the hash functions are good random functions, $\mathsf{E}(|\Psi_d|) = (1 - (1 - 1/2^{11})^{M_d}) \cdot 2^{11}$. For example, when $t_y = 32, |M_d| = 64$, the expected number of false positives among y's children in $\mathcal{G}_m$ is less than 1. The total expected number of false positives is approximately the sum of the expected numbers of false positives in y's children in $\mathcal{G}_m$ for all y in the sets $\{S_d\}_{0 \le d \le maxd}$. In subsection 3.3, we see that this scheme can already sustain DDoS attack from 50 distributed attacker sites, which is twice as high as the Fragment Marking scheme.

The computational complexity of this scheme is also much lower than the Fragment Marking scheme, $O(\sum_d |S_d| \cdot |\Psi_{d+1}|)$ instead of $O(\sum_d |S_d| \cdot |\Psi_{d+1}|^8)$. Also, given the same marking probability q, this scheme only needs one eighth of the packets as the Fragment Marking scheme to reconstruct the attack graph.

3.2 Advanced Marking Scheme II

Although the Advanced Marking Scheme I are more efficient and accurate than the Fragment Marking scheme in case of DDoS, it still gives false positives when there are more than about 60 distributed attacker sites. The reason is that 11-bit hash value is not sufficient to avoid collision when there are many routers at the same distance to the victim in the attack graph. In order to be more robust against larger scale DDoS, we further extend the scheme. In particular, instead of using just two hash functions, we use two sets of independent hash functions. The intuition is that the probability of any false positive a to have the same hash value as a router b for one hash function h_1 is $1/2^{11}$, and the probability of a to have the same hash values as b for m independent hash functions is $(1/2^{11})^m = 1/2^{11 \cdot m}$.

Note that a standard way to generate a set of 2^a independent hash functions $\{h_i(.)\}_i$ is to use one hash function, i.e. g,

Marking procedure at router R_i:
  for each packet P
    let u be a random number from $[0, 1)$
    if $u \le q$ then
      P.distance $\leftarrow 0$
      P.edge $\leftarrow h(R_i)$
    else
      if (P.distance == 0) then
        P.edge $\leftarrow$ P.edge $\oplus\ h'(R_i)$
      P.distance $\leftarrow$ P.distance + 1

Reconstruction procedure at victim v:
  let S_d be empty for $0 \le d \le$ *maxd*
  for each child R of v in $\mathcal{G}_m$
    if $h(R) \in \Psi_0$ then
      insert R into S_0
  for d:= 0 to *maxd* $- 1$
    for each y in S_d
      for each x in Ψ_{d+1}
        $z = x \oplus h'(y)$
          for each child u of y in $\mathcal{G}_m$
            if $h(u) = z$ then
              insert u into S_{d+1}
  output S_d for $0 \le d \le$ *maxd*

Figure 4. Advanced Marking Scheme I

and let $h_i(x) = g(\langle i, x \rangle)$, where i is an a-bit index, and $\langle _, _ \rangle$ represents bit concatination. Suppose we use 2^w independent hash functions in this scheme. Every time when a router decides to mark the packet, it would choose one of the 2^w hash functions for the encoding. And when the victim reconstructs the attack graph, it would need to know which hash function the router used to mark each individual packet.

One approach is to use some packet-specific data to determine which hash function the router should choose and to indicate to the victim which hash function the router has used. For example, for a packet P containing source IP address $\mathsf{sourceIP}$, the encoding of the router IP R_i could be $g(\langle f(\mathsf{sourceIP}), R_i \rangle)$, where f is a function mapping from 32 bits to w bits. $f(x)$ could be simply the first w bits of x, or a better solution, be another independent hash function. Thus, with packets containing different source IP addresses, the victim can hopefully get markings of each router with 2^w independent hash functions. Unfortunately this approach is not robust against the countermeasure of attackers. First, the attackers could simply use the same spoofed source IP address (this might be questionable since if the attackers use the same source IP address, the victim can easily block it.). Second, the attackers could carefully compute the source IP addresses such that they all hash into the same w bits. Since w is normally small for efficiency reason, the second countermeasure is practical. In this case, the packets marked by the same router will only be marked with hash values from one hash function, instead of a set of 2^w independent hash functions as mentioned before.

Therefore, we use another approach where we use an explicit *flag* field to indicate which hash function the router has used for the marking. In particular, we divide the overloaded

Cloudflare, Inc. Exhibit 1014 Page 5

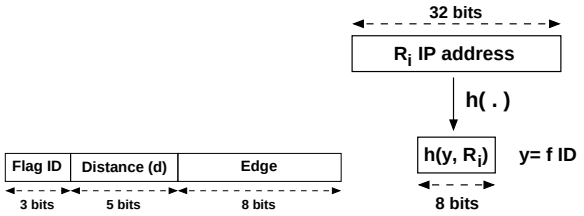


Figure 5. Encoding for Advanced Marking Scheme II

IP Identification field into a w-bit flag field, *fID*, a $(11-w)$-bit edge field, and a 5-bit distance field. Figure 5 shows an example of this approach for $w = 3$. With a given fID, the encoding of a router R_i is simply $h(\langle \mathsf{fID}, R_i \rangle)$. Thus different fIDs indicate different independent hash functions. When a router R_i decides to mark a packet, it chooses a random number x of w bits and write it in the flag field and use $g(\langle x, R_i \rangle)$ as its IP address encoding, as Figure 5 shows. The rest of the scheme is similar to the Advanced Marking scheme 1. Figure 6 shows a more detailed description of the scheme in the case of $w = 3$.

Advanced Marking Scheme II at router R_i:
  for each packet P
    let u be a random number from $[0, 1)$
    if $u \leq q$ then
      let x be a random integer from $[0, 7)$
      P.fID $\leftarrow x$
      P.distance $\leftarrow 0$
      P.edge $\leftarrow g(\langle x, R_i \rangle)$
    else
      if (P.distance == 0) then
        P.edge $\leftarrow$ P.edge $\oplus g'(\langle$P.fID, $R_i \rangle)$
      P.distance $\leftarrow$ P.distance + 1
Reconstruction procedure at victim v:
  let S_d be empty for $0 \leq d \leq$ *maxd*
  for each child R of v in $\mathcal{G}_m$
  let *count* = 0
    for l:= 0 to $2^w - 1$
      if $g(\langle l, R \rangle) \in \Psi_{0,l}$ then
        $count = count + 1$
    if $count > m$ then
      insert R into S_0
  for d:= 0 to *maxd* $- 1$
    for each y in S_d
      for each child u of y in $\mathcal{G}_m$
        let *count* = 0
        for l:= 0 to $2^w - 1$
          for each x in $\Psi_{d+1,l}$
            $z = x \oplus g'(\langle l, y \rangle)$
            if $g(\langle l, u \rangle) = z$ then
              $count = count + 1$; break
        if $count > m$ then
          insert u into S_{d+1}
  output S_d for $0 \leq d \leq$ *maxd*

Figure 6. Advanced Marking Scheme II

The reconstruction algorithm is similar to the Advanced Marking Scheme 1, except that here we use a threshold scheme. Recall that $\mathcal{G}_m$ denotes the map of upstream routers from the victim. Let's denote the set of unique edge segments marked with a distance d and flag ID l as $\Psi_{d,l}$. For a m-threshold scheme, a node u in $\mathcal{G}_m$ will only be considered as on an attack path if more than m of its hash values from the 2^w hash functions match the right markings in the attack packets. More details of the reconstruction procedure is described in figure 6.

Assume in a DDoS attack, let $|M_d|$ denotes the number of routers on the attack paths at distance d from the victim. Recall S_{d-1} denotes the set of routers at distance $d-1$ to the victim in the reconstructed attack graph. For every element y in S_{d-1}, let t_y denotes the in-degree of y (the number of y's children) in $\mathcal{G}_m$. Then in a 2^w-threshold scheme, the expected number of false positives among y's children is $t_y \cdot \prod_{1 \leq l \leq 2^w} \frac{\Psi_{d,l}}{2^{11-w}}$. Assume the hash functions are good random functions, $\mathsf{E}(|\Psi_{d,l}|) = (1-(1-1/2^{11-w})^{M_d}) \cdot 2^{11-w}$. For example, when $w = 3, t_y = 32, |M_d| = 128$, the expected number of false positives among y's children is less than 1. The total expected number of false positives is approximately the sum of the expected numbers of false positives in y's children in $\mathcal{G}_m$ for all y in the sets $\{S_d\}_{0 \leq d \leq maxd}$. In subsection 3.3, we see that this scheme can sustain DDoS attack from over 1500 distributed attacker sites.

The computational complexity of this scheme is still $O(\sum_d |S_d| \cdot |\Psi_{d+1}|)$ instead of $O(\sum_d |S_d| \cdot |\Psi_{d+1}|^8)$ in the Fragment Marking scheme. Although in the case of $w = 3$, this scheme with threshold $m > 7$ needs about the same number of packets as the Fragment Marking scheme given the same marking probability q, this scheme has the advantage that it can already start reconstructing the attack graph when it only receives a fraction of the packets as needed in the Fragment Marking scheme, and the more packets it receives, the attack graph simply becomes more accurate.

3.3 Simulation Results

To test the behavior of these Advanced Marking schemes in real settings, we conduct an experiment on simulated attacks using a real traceroute dataset obtained from AT&T [che99]. The traceroute dataset contains 709,310 distinct traceroute paths from a single source to 103,402 different destinations widely distributed over the entire Internet. In all the tests, we use the single source as the victim, and the whole traceroute dataset as the map of upstream routers from the victim. In each test, we randomly select a given number of destinations in the dataset as attackers. We then simulate the routers to mark the attack packets, and simulate the victim to reconstruct the attack graph using the markings in the packets. As indicated in the theoretical analysis, the number of false positives and computation time is related to the distribution of the number of rounters $|M_d|$ at a distance d, $0 \leq d \leq maxd$, in the attack graph. Even with the same number of attackers, the distribution of $\{|M_d\}_{0 \leq d \leq maxd}$ could be very different depending on the convergence of the attack paths. For most of the data points in the figures, we perform about 50–100

Cloudflare, Inc. Exhibit 1014 Page 6

inde

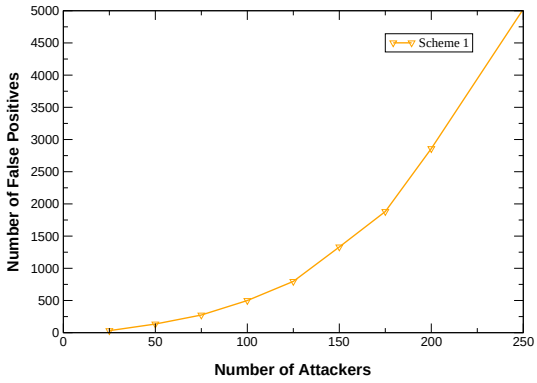


Figure 7. False Positives for Advanced Marking Scheme I

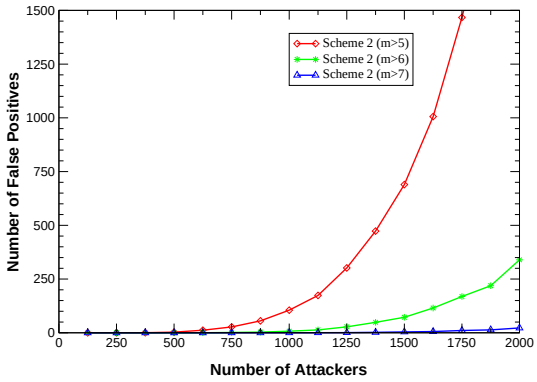


Figure 8. False Positives for Advanced Marking Scheme II

Figure 7 and figure 8 shows the number of false positives of the Advanced Marking Scheme I and the Advanced Marking Scheme II with $w = 3$ and three different thresholds $m > 5, m > 6$ and $m > 7$. The Advanced Marking Scheme I can sustain DDoS attacks from fewer than 50 distributed attacker sites, while the Advanced Marking Scheme II with threshold $m > 5$ can sustain 500 distributed attacker sites with very few false positives, and with threshold $m > 6$ can sustain 1000 attacker sites. Finally, the Advanced Marking Scheme II with threshold $m > 7$ can be robust against DDoS with even 1500 distributed attacker sites and only give about 20 false positives when there are 2000 attacker sites.

Figure 9 shows the time used to reconstruct the attack graph by the victim after the victim has received all the packets needed. The Advanced Marking Scheme I took substantially longer as the number of attackers increase because it has many more false positives. For the Advanced Marking Scheme II, all three different thresholds took less than 100 seconds to reconstruct the attack graph even when there are 2000 distributed attacker sites.

For comparison purpose, we also preformed similar simulation using the Fragment marking scheme, shown in figure 10 and 11. With only 20 attackers, the scheme already outputs over 100 false positives and take more than a day to reconstruct the attack graph. With 25 attackers, the scheme outputs thousands of false positives and cannot terminate within a week. Note that our timing information is based on a highly optimized implementation of the Fragment Marking Scheme running on a 500 MHz Pentium III Linux workstation. So for simulations with more than 20 attackers, we compute the expected number of false positives and expected computation overhead[2] using the formulas in our theoretical analysis (for details, see appendix A). The main reason for the dramatic increase of the number of false positives in the Fragment Marking Scheme is because the accumulative explosion effect – more false positives closer to the victim causes many more false positives further to the victim due to the reconstruction process.

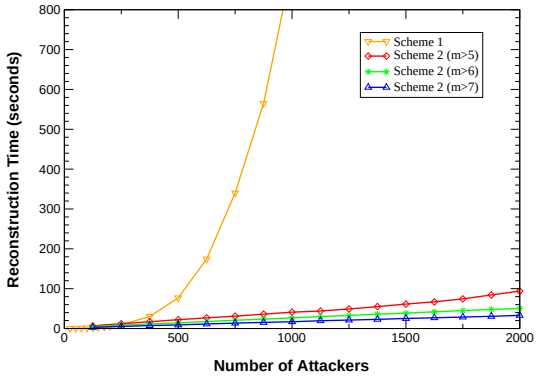


Figure 9. Computation Overhead for Advanced Marking Schemes

We also tested the nubmer of packets required to reconstruct the attack graph. Figure 12 and 13 show the simulation result of the number of packets required to reconstruct paths of varying length with 95% probability in presence of only one attacker for the Fragment Marking Scheme and our Advanced Marking Schemes with $w = 3$. Each data point is average of over 100 independent random tests with given length value. The marking probability $q = 4\%$ in figure 12, and $q = 1\%$ in figure 13. As described in section 3.2, the Fragment Marking Scheme and the Advanced Marking Scheme II with $w = 3$ and threshold $m > 7$ require the same number of packets to reconstruct the attack paths, while the Advanced Marking Scheme II with $w = 3$ and threshold $m > 6$ and $m > 5$ require substantially fewer packets for the reconstruction. Hence the Advanced Marking Scheme has the advantage that it can already start reconstructing the attack graph when it only receives a fraction of the packets as needed in the Frag-

[2]The expected computation overhead is computed by first computing the expected number of hash computations needed and then assume we can do 6 million hash computations per second as indicated in the experiments of fast hash computations.

Cloudflare, Inc. Exhibit 1014 Page 7

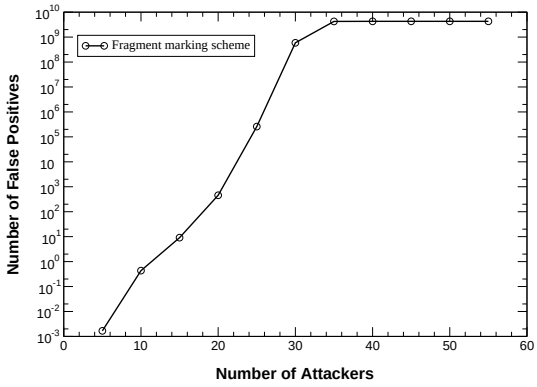


Figure 10. False Positives for Fragment Marking Scheme

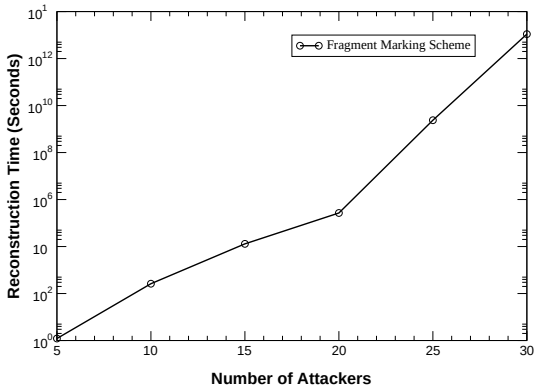


Figure 11. Computation Overhead for Fragment Marking Scheme

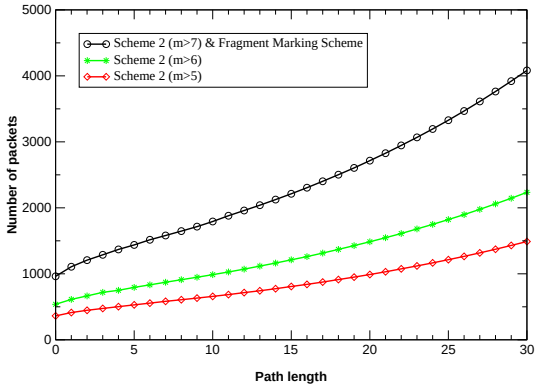


Figure 12. Number of Packets Required for Reconstruction ($q = 4\%$)

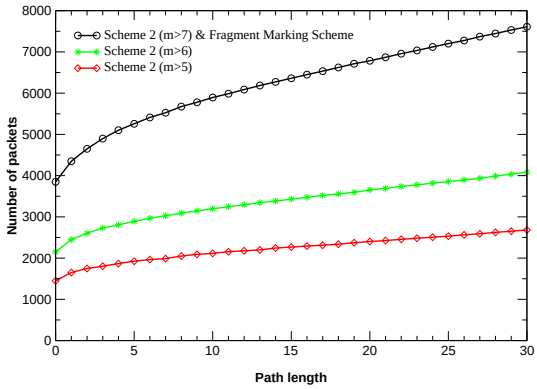


Figure 13. Number of Packets Required for Reconstruction ($q = 1\%$)

ment Marking scheme, and the more packets it receives, the attack graph simply becomes more accurate.

4 Authenticated Marking Scheme

A fundamental shortcoming of the advanced marking schemes is that the packet markings are not authenticated. Consequently, a compromised router on the attack path could forge the markings of its upstream routers. Moreover, the compromised router could forge the markings according to the precise probability distribution, preventing the victim from detecting and determining the compromised router by analyzing the marking distribution. To solve this problem, we need a mechanism to authenticate the packet marking. A straightforward way to authenticate the marking of packets is to have the router digitally sign the marking it adds to the packet. However, digital signatures have two major disadvantages. First, they are very expensive to compute (a 500 MHz Pentium can only compute on the order of 100 1024-bit RSA signatures per second). Secondly, the space overhead is large (128 bytes for a 1024-bit RSA signature).

We propose a much more efficient technique to authenticate the packet marking, the Authenticated Marking Scheme. This technique only uses one cryptographic MAC (Message Authentication Code) computation per marking, which is orders of magnitude more efficient to compute (i.e., HMAC-MD5 is 10^3 up to 10^4 times more efficient than 1024-bit RSA signing) and can be adapted so it only requires the 16-bit overloaded IP identification field for storage.

4.1 Step 1: Authentication with a MAC

Message Authentication Codes (MAC) such as HMAC [KBC97] are commonly used for two-party message authentication. Two parties can share a secret key K. When party A sends a message M to party B, A appends the message with the MAC of M using key K. When B receives the message, it can check the validity of the MAC. A well-designed MAC guarantees that nobody can forge a MAC of a message without knowing the key. MAC computation is very efficient, e.g. a fast workstation can compute on the order of one million 8-byte HMAC-MD5 per second.

Cloudflare, Inc. Exhibit 1014 Page 8

Let f denote a MAC function and f_K denotes the MAC function using key K. If we assume that each router R_i shares a unique secret key K_i with the victim, then instead of using hash functions to generate the encoding of a router's IP address, R_i can apply a MAC function to its IP address and some packet-specific information with K_i. Because a compromised router still does not know the secret keys of other uncompromised routers, it cannot forge markings of other uncompromised routers. The packet-specific information is necessary to prevent a replay attack, because otherwise, a compromised router can forge other routers markings simply by copying their marking into other packets. We could use the entire packet content in the MAC computation, i.e. encode R_i as $f_{K_i}(\langle \mathsf{P}, R_i \rangle)$. But for efficiency, it might also be sufficient to just use the source and destination IP addresses in the packet, i.e. encode R_i as $f_{K_i}(\langle \mathsf{sourceIP}, \mathsf{destinationIP}, R_i \rangle)$. In this case, a compromised router might still be able to forge a marking in a packet by using the same source IP address, but in this case, the victim can block traffic coming from this source IP address and hence the attackers would not succeed in the denial-of-service attack. (Also the extended scheme in step 2 can reduce the possible number of source IP addresses that the compromised router could use to replay.) Besides the change of using a MAC function with secret keys instead of publicly available hash functions, the marking and reconstruction procedure is similar to the Advanced Marking schemes.

4.2 Step 2: Using Time-Released Key Chains

Although Step 1 can provide router authentication, it is obviously impractical because it requires each router to share a secret key with each potential victim. To solve this problem, we extend the scheme by using the time-released keys authentication scheme. A similar scheme was proposed by Perrig et al. for authentication of multicast data streams over lossy channel [PCTS00].

The basic idea is that each router R_i first generates a sequence of secret keys, $\{K_{j,i}\}$ where each key $K_{j,i}$ is an element of a hash chain. By successively applying a one-way function g (e.g. a cryptographic hash function such as MD5 [Riv92]) to a randomly selected seed, $K_{N,i}$, we can obtain a chain of keys, $K_{j,i} = g(K_{j+1,i})$. Because g is a one-way function, anybody can compute forward, e.g. compute $K_{0,i}, \ldots, K_{j,i}$ given $K_{j+1,i}$, but nobody other than R_i can compute backward, e.g. compute $K_{j+1,i}$ given only $K_{0,i}, \ldots, K_{j,i}$.

R_i then commits to the secret key sequence through a standard commitment protocol, e.g. by signing the first key of the chain $K_{0,i}$ with its private key, and publish the commitment out of band, e.g. by posting it on a web site. We assume that each router has a certified public key.

The time is divided into intervals (as shown later, the time interval needs to be sufficiently long, e.g. on the order of ten seconds). Each router R_i then associates its key sequence with the sequence of the time interval, with one key per time interval. In time interval t, the router R_i uses the key of the current interval, $K_{t,i}$, to mark packets in that interval. The marking scheme is the same as in Step 1, except that instead of using a shared secret key between R_i and the victim to compute the marking, the router uses $K_{t,i}$ as the key to compute the MAC. R_i will then reveal the key $K_{t,i}$ after a delay of δ_r after the end of the time interval t. The disclosure time delay δ_r is on the order of few time intervals, as long as it is greater than any reasonable round trip time on the Internet plus the dispersion[3]. The disclosure of the keys can be done out of band, e.g. published on a web-site.

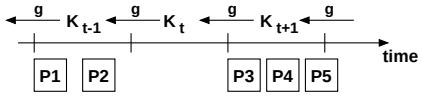


Figure 14. Authenticated Marking Using a Time-Released Key Chain at a Router.

Figure 14 shows an example of using time-released secret key chain for the authenticated marking scheme. Note that because of the use of the key chain, the victim only needs to download the keys of the routers for the latest time interval and then it is able to compute all the keys for previous time intervals.

When the victim receives the marked packets, it saves the arrival time for each packet. Note that the victim and the routers need an approximate time synchronization. For the purpose of this approach, the time only needs to be loosely synchronized, e.g. the dispersion can be on the order of multiple seconds. This level of approximate time synchronization is easy to achieve in practice. Please refer to [Mil92, PCTS00] for more details on time synchronization.

Before reconstructing the attack graph, the victim downloads the disclosed keys of the routers. Note that the victim does not need to download the keys of all routers at once. It could start from its nearest routers to routers further away as necessary in the reconstruction process. For each marked packet, it first determines the sending time interval of the packet using its arrival time. Suppose the arrival time of the packet is T_a, the time synchronization between the victim and the router is $\pm\delta_s$, and the maximum transmission delay of the packet is δ_d. Thus, the correct sending time of the packet T_s is bounded as $T_a - \delta_s - \delta_d < T_s < T_a + \delta_s$. Therefore, if the length of the time interval is substantially longer than $2\delta_s + \delta_d$, the victim could determine the sending time interval of the marked packet with high probability. After determining the sending time interval of the marked packets, the victim can associate the right keys used to compute the MACs with the packets. It can then use a similar reconstruction algorithm as the Advanced Marking Schemes to reconstruct the attack graph.

[3]The dispersion is the maximum offset of the time synchronization between the router and the victim [Mil92]

Cloudflare, Inc. Exhibit 1014 Page 9

Note that because we use the source IP address and different keys over different time intervals in the computation of the encoding of router IP addresses, it has the same effect as using a set of independent hash functions as described before. So it might not be necessary to use an explicit flag field as in the Advanced Marking Scheme 2 if we mainly consider DDoS attacks that last at least multiple minutes.

5 Discussion

5.1 Mapping Upstream Routers

In previous sections, we show that the Advanced Marking Schemes and the Authenticated Marking Scheme are very efficient and accurate even in presence of large distributed denial-of-service. But these marking schemes rely on the assumption that the victim has a map of upstream routers. In this subsection, we show that this assumption is reasonable and practical.

First, it is easy to obtain such a map of upstream routers for a victim. There are standard tools such as a tool based on traceroute from AT&T [che99] and Skitter from CAIDA [CAI]. These tools can obtain the map of upstream routers from the victim to over 100,000 destinations per day with a very non-intrusive manner.

Second, such a map does not need to have high accuracy and does not need to be very recent, as long as the attack graph is contained in this map. Furthermore, even if the victim does not have such a map before it is attacked by DDoS, it can obtain the map after being attacked and the result will mostly not be affected.

Finally, in the real deployment of this protocol, we can also build in exploration protocols to support incremental deployment. Thus the victim can get the map of upstream routers which implement the advanced marking protocol and the authenticated marking protocol.

5.2 Related Work

Researchers have proposed various schemes to address the IP traceback problem. Unfortunately they are mostly inefficient or ineffective and not robust against DDoS. Ferguson and Senie proposed ingress filtering where each router blocks packets that arrive with illegitimate source addresses [FS98]. This approach requires the router to have sufficient power to verify the IP address of each packet and to have sufficient knowledge to distinguish between legitimate and illegitimate addresses. Also the effectiveness of the approach depends of widespread deployment. Burch and Cheswick proposed controlled flooding [BC99]. In this approach, the victim floods some network selectively during the attack to check the correlation of the flooding with the attack and gather information about the sources of the attacks from the correlation. This approach is only applicable during on-going attacks, introduces large overhead and cannot deal with DDoS. Sager [Sag98] and Stone [Sto99] propose logging on routers. This approach has high overhead of storage and processing. All these previous work does not consider the issue of router authentication.

Bellovin [Bel00] proposed to use ICMP messages for authenticated IP marking. In this scheme, each router probabilistically generates a copy of a packet signed with its private key when forwarding the packet. It requires the routers to create new packets and compute expensive digital signatures, and hence, more expensive than our Authenticated Marking Scheme. Given the marking probability q, when $q < 0.005$, the two schemes would need about the same number packets to reconstruct the attacking paths. In our Authenticated Marking Scheme, in the case that a router does not mark a packet, it only needs to increment the distance field most of the time. And as proposed in [SWKA00], the distance field will be aligned with the TTL field such that the router even need to do less work in this case since the checksum will not need to be changed. In the case that a router marks a packet, it only does a MAC computation which is significantly less work than computing a signature and creating a new packet. Also, if q is low, the marking procedure could be implemented in software.

6 Conclusion

In this paper, we have presented two new schemes, the Advanced Marking Scheme and the Authenticated Marking Scheme, which allow the victim to traceback the approximate origin of spoofed IP packets. Our techniques have very low network and router overhead and support incremental deployment. Comparing with previous work, our techniques have significantly higher precision (lower false positive rate) and lower computation overhead for the victim to reconstruct the attack paths under large scale distributed denial-of-service attacks. Furthermore the Authenticated Marking Scheme provides efficient authentication of routers' markings such that even a compromised router cannot forge or tamper markings from other uncompromised routers.

Acknowledgment

We would like to thank Eric Brewer and Doug Tygar for their encouragement for this work.

References

[BC99] Hal Burch and Bill Cheswick. Tracing anonymous packets to their approximate source. Unpublished paper, December 1999.

[Bel00] Steve Bellovin. The icmp traceback message. `http://www.research.att.com/~smb`, 2000.

[CAI] `http://www.caida.org`.

Cloudflare, Inc. Exhibit 1014 Page 10

[CC97] Robert Carter and Mark Crovella. Dynamic server selection using dynamic path characterization in wide-area networks. In *Proceedings of the 1997 IEEE INFOCOM Conference*, April 1997.

[che99] Internet mapping. `http://cm.bell-labs.com/who/ches/map/dbs/index.html`, 1999.

[FS98] P. Ferguson and D. Senie. Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing. RFC 2267, January 1998.

[How98] John Howard. *An Analysis of Security Incidents on the Internet*. PhD thesis, Carnegie Mellon University, 1998.

[KBC97] H. Krawczyk, M. Bellare, and R. Canetti. Hmac: Keyed-hashing for message authentication. Internet RFC 2104, February 1997.

[Mea99] Catherine Meadows. A formal framework and evaluation method for network denial of service. In *Proceedings of the 1999 IEEE Computer Security Foundations Workshop*, June 1999.

[Mil92] David L. Mills. Network Time Protocol (Version 3) Specification, Implementation and Analysis. Internet Request for Comments, March 1992. RFC 1305.

[PCTS00] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Xiaodong Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, May 2000.

[Riv92] R. L. Rivest. The MD5 message digest algorithm. RFC 1321, Internet Activities Board, Internet Privacy Task Force, April 1992, 1992.

[Sag98] Glenn Sager. Security fun with ocxmon and cflowd. Presentation at the Internet 2 Working Group, November 1998.

[SP99] Oliver Spatscheck and Larry Peterson. Defending against denial of service attacks in scout. In *Proceedings of the 1999 USENIX/ACM Symposium on Operating System Design and Implementation*, February 1999.

[Sto99] Robert Stone. Centertrack: An ip overlay network for tracking dos floods. Unpublished, October 1999.

[SWKA00] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for ip traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, August 2000. An early version of the paper appeared as techreport UW-CSE-00-02-01 available at: `http://www.cs.washington.edu/homes/savage/traceback.html`.

[Sys97] Cisco Systems. Configuring tcp intercept (prevent denial-of-service attacks). Cisco IOS Documentation, 1997.

[SZ99] Ion Stoica and Hui Zhang. Providing guaranteed services without per flow management. In *SIGCOMM'99*, pages 81–94, 1999.

[Tea00] Computer Emergency Response Team. Cert advisory ca-2000-01 denial-of-service developments. `http://www.cert.org/advisories/CA-2000-01.html`, 2000.

[TR00] Wolfgang Theilmann and Kurt Rothermel. Dynamic distance maps of the internet. In *Proceedings of the 2000 IEEE INFOCOM Conference*, March 2000.

A Analyzing the Limitation for DDoS in the Fragment Marking Scheme

The paper [SWKA00] mainly considers denial-of-service with a single attacker site. In this section, we analyze and illustrate that the Fragment Marking scheme is inefficient and inaccurate under even a small scale of DDoS.

In the Fragment Marking scheme, each attack packet contains the 16-bit marking block including a 5-bit *distance* field, a 3-bit *fragment ID* field, and a 8-bit *edge fragment* field. Each router's IP address is encoded using eight 11-bit fragments and each packet will probabilistically contain one of the eight fragments from the router who marked the packet. We denote the set of unique edge fragments marked with a distance d and fragment ID f as $\Psi_{d,f}$. Because for each distance d, in the eight sets $\Psi_{d,0}, \ldots, \Psi_{d,7}$, the victim cannot distinguish which eight fragments are from the encoding of the same router, in order to reconstruct the attack graph, the victim needs to consider all possible ordered combination of the eight sets $\Psi_{d,0}, \ldots, \Psi_{d,7}$ and check which combinations have the right format (by checking that the hash value of the odd bits match the even bits). We denotes the set of these combinations as C_d. Clearly, $|C_d| = \prod_{0 \le f \le 7} |\Psi_{d,f}|$. In case of DDoS, $|\Psi_{d,f}|$ could be quite high. Thus, with presence of multiple attacker sites, the Fragment Marking scheme severely suffer from the following two main problems:

- High computation overhead

  Assume in the reconstructed attack graph, the number of distinct routers at distance d is $|S_d|$. Then to reconstruct the routers at distance $d+1$, the victim XORs each element x in C_{d+1} with each element y in S_d and computes

Cloudflare, Inc. Exhibit 1014 Page 11

$z = x \oplus y$ and checks whether z has the right format. Denote the set of the XOR results as Γ_{d+1}, the number of combinations to be checked is

$$|\Gamma_{d+1}| = |S_d| \times |C_{d+1}| = |S_d| \times \prod_{0 \le f \le 7} |\Psi_{d+1,f}|.$$

So the total number of combinations to be checked for all the distances is

$$|\Gamma| \quad = \sum_{0 \le d \le maxd} (|S_{d-1}| \times \prod_{0 \le f \le 7} |\Psi_{d,f}|),$$

using convention $|S_{-1}| = 1$. It requires at least one hash computation to check one combination. As we show in the experiments in subsection 3.3, even using a fast hash function implementation which can compute 6 million 32-bit hash value per second, it can take days to check all the combinations in a DDoS with 25 attackers.

- Large number of false positives

  For each such z in the set Γ, the probability that it is a valid IP encoding is $1/2^{32}$ when z is not on the attack paths, because the hash value is 32 bits. So the expected number of elements in Γ that are valid IP encoding is $|\Gamma|/2^{32}$, denoted as α. Because an IP address is 32 bits, so the expected number of false positives is

  $$\mathsf{E}[\text{false positives}] = (1 - (1 - 1/2^{32})^{\alpha}) \cdot 2^{32}.$$

  When $\alpha \ll 2^{32}$, $\mathsf{E}[\text{false positives}] \doteq \alpha = |\Gamma|/2^{32}$. As we show in the experiments in subsection 3.3, even for a DDoS with 25 attackers, the reconstruction can result in thousands of false positives.

Cloudflare, Inc. Exhibit 1014 Page 12