FIT: Fast Internet Traceback

Abraham Yaar Adrian Perrig Dawn Song
Carnegie Mellon University

***Abstract*— Traceback mechanisms are a critical part of the defense against IP spoofing and DoS attacks, as well as being of forensic value to law enforcement. Currently proposed IP traceback mechanisms are inadequate to address the traceback problem for the following reasons: they require DDoS victims to gather thousands of packets to reconstruct a single attack path; they do not scale to large scale Distributed DoS attacks; and they do not support incremental deployment.**

We propose Fast Internet Traceback (FIT), a new packet marking approach that significantly improves IP traceback in several dimensions: (1) victims can identify attack paths with high probability after receiving only tens of packets, a reduction of 1–3 orders of magnitude compared to previous packet marking schemes; (2) FIT performs well even in the presence of legacy routers, allowing every FIT-enabled router in path to be identified; and (3) FIT scales to large distributed attacks with thousands of attackers. Compared with previous packet marking schemes, FIT represents a step forward in performance and deployability.

I. Introduction

E-CRIME is on the rise. Estimates for the cost of worm and virus epidemics, and Distributed Denial of Service (DDoS) attacks are often on the order of several billions of dollars. Even if the true cost is one order of magnitude smaller, we would still need to address these problems, especially since the frequency and severity of these attacks are increasing. One of the most severe DDoS attacks, against the Domain Name System (DNS) in January 2003, proves that even the critical infrastructure of the Internet is potentially vulnerable. More recently, attackers target victims for political or economic reasons—SCO, RIAA, and anti-spam blacklist servers have all been the subject of such attacks. Lastly, attacks are sometimes used to extort money from their victims. Attackers request “protection” money to stop the attack or guarantee that it will not be repeated. Twenty UK betting sites, the e-commerce firm “2Checkout”, and e-book distributors were all recent victims of attacks that occurred in conjunction with extortion attempts. Many attacks, including large scale DDoS attacks, use spoofed source IP addresses to protect the perpetrators or their assets.

Unfortunately, the current Internet infrastructure does not provide mechanisms for e-crime victims to defend themselves against attackers who use IP spoofing. Law enforcement could greatly benefit from IP traceback methods, particularly in the cases involving extortion. Finally, traceback mechanisms are a first step in providing automated packet filtering at routers in order to block an attack’s origin.

To effectively provide the above benefits and be applicable in an Internet environment, a traceback mechanism must have the following general properties:

IncDep: A traceback mechanism should function even when only partially deployed across routers in the Internet.

RtrChg: A traceback mechanism should only require a small hardware change on routers.

FewPkt: A traceback mechanism should allow the victim to identify the attack path after only a small number of packets.

Scale: A traceback mechanism should scale to a large number of attackers while maintaining accuracy (as measured by incorrect implication of non-attacking endhosts and routers (false positives), and the failure to identify true attack endhosts and routers (false negatives)).

Local: A traceback mechanism should allow an attack victim to perform traceback locally, without communicating with any router or ISP.

Table I shows that none of the major traceback mechanisms provides all of the above properties, besides Fast Internet Traceback (FIT). A detailed discussion of prior traceback proposals is in Section II.

Contributions In this paper, we propose FIT, a new probabilistic packet-marking approach for IP traceback that achieves much stronger properties than previous schemes. We list a summary of the contributions of this paper:

- The FIT traceback scheme is very efficient. In contrast to previous work, FIT simultaneously achieves all the following properties: tens of packets to trace an attack path, scales to thousands of distributed attackers, incrementally deployable, and no per-flow or per-packet state required at routers.
- FIT provides the unique property that the victim can detect the presence of legacy routers on the attack path. Previous traceback mechanisms either fail completely when large numbers of legacy routers are present, or misrepresent reconstructed router locations along the path by counting only traceback enabled routers in their distance measurement. FIT

This research was supported in part by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office, and by a gift from Matsushita Electric Works. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, Carnegie Mellon University, Matsushita Electric Works, or the U.S. Government or any of its agencies.

Mechanism	IncDep	RtrChg	FewPkt	Scale	Local
Burch & Cheswick [3]	✔	✔			
FMS [15]	✔	✔			✔
AMS [17]		✔		✔	✔
iTrace [2]	✔	✔		✔	✔
Goodrich [8]	✔	✔			✔
Algebraic Traceback [5]	✔	✔			✔
SPIE [12], [16]			✔	✔	
FIT (this paper)	✔	✔	✔	✔	✔

TABLE I

COMPARISON OF PROPOSED TRACEBACK MECHANISMS. THE SYMBOL ✔ MEANS THAT THE MECHANISM PROVIDES THE PROPERTY.

properly identifies the distance in router hops from the victim regardless of whether intervening routers are traceback-enabled or not.

- FIT achieves these strong properties due to several novel techniques. First, in FIT, the victim uses an upstream-router map to provide its fast (in the number of packets received) attack path reconstruction. Although previous mechanisms have used this approach [17], FIT does not obtain the router map through out-of-band tools, such as traceroute. Rather, FIT can use the same router markings for constructing the upstream router map as it does for reconstructing traffic paths during an attack. Second, FIT uses a new mechanism for conveying the distance between a marking router and victim which uses only a single bit. Finally, because it accurately detects the distance – including both traceback-enabled and legacy router hops – of a marking router from the victim, FIT can use node sampling instead of edge sampling used in previous schemes; which allows for a significant reduction in the number of packets needed for path reconstruction during attacks.

Outline. The remainder of the paper is organized as follows: In Section II we present previous and related work in IP Traceback and discuss the tradeoffs of other methods. In Section III we describe FIT in detail and evaluate it in Section IV. We discuss some of the ramifications and future directions for FIT in Section V and we conclude in Section VI.

II. PREVIOUS WORK ON IP TRACEBACK

In this section, we first present an overview of the related work in IP traceback, and then we analyze the specific tradeoffs of these approaches.

A. *Overview of previous traceback mechanisms.*

The importance of IP traceback has prompted many researchers to work on this topic [1], [2], [5], [8], [11], [12], [15], [16], [17]. We review these efforts in chronological order.

Burch and Cheswick introduce the concept of network traceback. They identify attack paths by selectively flooding network links and monitoring the changes caused in attack traffic [3].

Savage et al. propose the Fragment Marking Scheme (FMS) for IP traceback [15]. They suggest that routers probabilistically mark the 16 bit IP identification field, and that the receiver reconstructs the IP addresses of routers on the attack path using these markings.

Bellovin et al. develop iTrace [2]. In iTrace, routers probabilistically send a message to either the source or destination IP address of a packet, indicating the IP address of the router. This approach does not alter packets in-flight and victims can also detect attackers that use *reflectors* to hide their presence [13], however, it does generate additional traffic.

Goodrich presents a marking scheme that marks nodes instead of links into packets [8]. Because this approach does not use a distance field, it has issues with attack graph reconstruction and does not scale to a large number of attackers.

Dean et al. suggest algebraic traceback, an algorithm to encode a router's IP address as a polynomial in the IP identification field [5]. We show in the next subsection that it does not scale to large number of attackers.

Adler presents a theoretical analysis of traceback, presenting a one-bit marking scheme [1]. This work is primarily of theoretical interest, and does not scale to large numbers of attackers.

Snoeren et al. propose SPIE, a mechanism using router state to track the path of a single packet [16]. The main advantage of SPIE is that it enables a victim to trace back a single packet by querying the router state of upstream routers, however, it does require routers to keep a large amount of state. Li et al. have further developed their approach, lowering the required router state, at the expense of a large communication overhead for traceback [12].

B. *Requirements for IP Traceback mechanisms and Analysis of Previous Approaches*

We now discuss the properties of an ideal traceback mechanism, and argue why previously proposed mechanisms do not achieve them.

To be viable for forensics and DDoS defense, a traceback approach needs to provide incremental benefits even when deployed on a small number of routers, should require only a small hardware change and minor computation overhead on the router, should require only a few packets to traceback to an attacker, should scale to large DDoS attacks with few false positive and false negatives, requires a small overhead on the victim for traceback, and enables the victim to perform traceback locally without relying on the communication infrastructure that is under attack.

We now discuss these properties in more detail and describe how previous traceback approaches achieve them. Table I shows a summary of properties and lists whether a scheme achieves it.

IncDep. Enabling incremental deployment is a very important requirement for any traceback mechanism. If a traceback algorithm does not provide benefits for incremental deployment, an ISP would have no incentive to start deployment. It is unrealistic to assume that after a “flag day”, 90% of all routers in the Internet will deploy traceback; it is more realistic that we will initially see 10–20% deployment which later reaches 50% deployment.

Unfortunately, most previous IP traceback mechanisms do not provide strong properties for incremental deployment. For example, the hash-based IP traceback mechanisms [12], [16] do not work well if only a small number of routers implement them. Consider the case where 50% of the routers implement the SPIE mechanism [16]. Let’s consider (very conservatively), that a router has 10 neighboring routers on average. With the SPIE mechanism, we find that a given router forwarded an attack packet, and we attempt to find out from which neighboring router it came from. Thus, we need to contact the 9 neighboring routers which potentially forwarded the packet (we do not need to query the next-hop router towards the victim). Let’s assume that 5 of the neighboring routers implement SPIE, but that 4 do not implement it. Besides the 5 SPIE-enabled routers, we also need to contact all neighbors of the 4 legacy routers, about 40 additional routers. However, 20 of those routers are legacy routers themselves, so we need to contact all of their neighbors as well. It is clear that this approach scales poorly if an attack path traverses several legacy routers.

The AMS approach [17] suffers from a similar problem. The upstream map of routers used by AMS is gathered using the traceroute tool, which does not distinguish between AMS-enabled and legacy routers. However, the AMS distance field only counts hops of AMS-enabled routers, which leads to the following problem. Assuming the victim has identified a router at distance x, when receiving an edge marking from distance $x+1$, the victim will have to test the IP addresses of all the routers at distances greater than x (rather than just those at distance $x+1$) because the edge between two AMS-enabled routers may traverse several non-marking legacy routers. This effect will lead to an increase in the false-positive rate of the scheme, particularly with high percentages of legacy routers present. FIT uses a node sampling mechanism that corrects this issue.

RtrChg. Deployment is linked with small router change and low router overhead. If a traceback mechanism only requires a minimal hardware change and has a negligible overhead for packet forwarding, it is more likely to be accepted by router manufacturers and eventually reach ISPs.

The SPIE traceback approach requires a multi-byte hash over the header and part of the payload of each incoming packet, as well as a large amount (approximately 1 GB) of memory to store its Bloom-filters. Although this can be done with the addition of dedicated hardware, it may affect hardware cost and will certainly require a non-negligible architectural change in those routers that implement it.

FewPkt. Complete traceback using only a small number of packets is especially useful for forensics. So far, only the SPIE mechanism enables single-packet traceback, and all other traceback approaches require on the order of thousands [3], [5], [8], [12], [15], [17], or tens of thousands [2] of packets. FIT can trace a single attacker using only tens of packets.

Scale. A viable traceback approach should be able to scale to large attacks, and enable traceback to tens of thousands of attackers with only a small number of false positives and false negatives. Unfortunately, most mechanisms do not scale well to large numbers of attackers because their false positive rate becomes prohibitively large [3], [8], [15]. The algebraic traceback scheme does not scale well either, as the number of packets N required from each attacker for reconstruction is linearly dependent on the number of attackers n (where p is the marking probability): $N > 5n/p^2$ [5]. Since we have n attackers, the total number of packets received is $5n^2/p^2$.[1] FIT is presented here with a baseline reconstruction scheme which scales to thousands of attackers. In Section V we present preliminary techniques to further improve FIT’s scalability.

Low overhead for attack path reconstruction. The reconstruction of the attack path should be efficient for the victim. The algebraic traceback scheme requires $O(m^{2.5})$ computations for reconstruction (where m is the number of fragments collected) [5]. The approach by Burch and Cheswick would require substantial network resources to send the additional packets for high-bandwidth network links [3], which does not satisfy our requirement for low overhead on the victim. Similarly, the approach by Li et al. has a high bandwidth overhead for the victim, as the victim needs to send about 1–10

[1] These are estimates based on formulas in their paper [5], however, they do not provide experiments or simulation results for real attack path reconstruction.

Mbytes of information to each router on the attack path for traceback [12].

Local. Enabling the victim to perform traceback locally is a desirable property because the victim may not be able to rely on the communication infrastructure that is under attack to perform traceback. Burch and Cheswick [3], and SPIE [12], [16] require the victim to use the network to perform traceback while the attack is happening.

Other shortcomings. In a *pollution attack*, the attacker sends malicious fragments that interfere with path reconstruction [10]. Since the traceback approach by Goodrich does not use a distance field, it is susceptible to a pollution attack, because the victim cannot distinguish between fragments generated by an attacker and those generated by marking routers. For example, if our marking probability is $q = 0.04$, the probability that packets arrive unmarked by an attacker at distance 12 hops is $(1 - q)^{1}2 = 0.96^{1}2 = 0.61$. Thus, almost two out of three markings created by the attacker arrive at the victim, which can prevent successful reconstruction.

With 100% deployment, other traceback schemes are less susceptible to pollution attacks because their distance fields allow marking injection only at distances greater than the closest attacker. However, large numbers of legacy routers reduce this immunity because distance is only counted in terms of marking-enabled routers. FIT has an arguably stronger resistance to pollution attacks, even under partial deployment, because it counts distance in terms of legacy and marking-enabled routers under most circumstances.

III. FIT: Fast Internet Traceback

Figure 1 shows the notation we use in this paper. In this section, we first present an overview of FIT, then we describe in detail how packet marking works, and how we can calculate the marking distance based on a single additional distance bit, followed by a detailed discussion on how FIT map and path reconstruction work.

A. FIT Overview

The FIT traceback mechanism is in the family of PPM (Probabilistic Packet Marking) traceback schemes [15], and consists of two major parts: a packet marking scheme to be deployed at routers, and map and path reconstruction algorithms used by endhosts receiving the packet markings.

In FIT, an attack victim is assumed to have constructed a map of upstream routers and their IP addresses using packet markings received before the attack itself occurs (we explain how this is achieved in Section III-D). Routers mark the 16-bit IP ID field of certain forwarded packets.[2] FIT packet markings contain three elements: a fragment of the hash of the marking router's IP address, the number of the hash fragment marked in the packet, and a distance field. Based on the distance field and the TTL of a given packet, the attack victim can determine from how many hops away the marking is generated. The victim uses the hash fragments and distance calculation from the markings in the malicious packets in conjunction with its router map to identify a candidate set of marking routers. After a number of different hash fragments matching a particular router arrive at the victim, that router is added to the reconstructed attack path.

b_{fnum}	Size of the fragment number field in bits
b_{frag}	Size of each fragment in bits, $b_{frag} = 15 - b_{fnum}$
c	Bit replacement for the 5 LSB of the TTL
n	Total number of fragments, $n = 2^{b_{fnum}}$
n_{map}	Number of unique fragments needed for single IP address map reconstruction
n_{path}	Number of unique fragments needed for single IP address path reconstruction
P.*dist_bit*	The distance bit in packet P
P.*frag_num*	The fragment number in packet P
P.*fragment*	The hash fragment in packet P
q	Marking probability
$TTL_{[0]}$	Least significant bit (LSB) of the TTL
$TTL_{[5]}$	Sixth bit of the TTL
$TTL_{[4..0]}$	The five least significant bits of the TTL
$H(IP)$	Compute a cryptographic hash function on the IP address, e.g., $SHA-1(IP)$
$b\|c$	Concatenation of the values b and c

Fig. 1. Notation we use in this paper.

Although FIT is superficially similar to the AMS traceback scheme (both use upstream router maps and packet markings with the fragment/number/distance format) FIT employs novel marking and reconstruction algorithms which dramatically improve its performance and make it a more viable traceback mechanism.

First, FIT allows the attack victim to generate the upstream router map using packet markings rather than the traceroute tool used in AMS. Traceroute generated maps have two serious deficiencies: they are inaccurate in the presence of asymmetric paths,[3] and they cause increased false positives because they do not distinguish between legacy and marking-enabled routers (as we discuss in Section II-B).

Second, the FIT marking mechanism uses node sampling instead of the commonly used edge sampling [5], [15], [17], greatly reducing the number of false positives and the number of packets required for attack path reconstruction.

[2] Many other packet marking schemes also use the IP ID field [5], [8], [15], [17] to hold their markings. We discuss how packet marking and IP fragmentation can coexist in an earlier work [18].

[3] Traceroute returns the path of packets from the victim to potential attackers, which may be different from the path of packets from potential attackers to the victim.

Third, FIT uses only 1-bit in the IP ID field to mark the distance from the victim at which the packet was marked. This allows 4 extra bits (previous traceback schemes used a 5 bit field) to be used for hash fragment marks, which both greatly reduces false positives (allowing FIT to scale to greater numbers of attackers) and increases the effective marking probability (we explain this in Section III-C) allowing FIT to traceback using fewer packets. In the following sections, we describe FIT in detail.

B. Packet Marking

In the FIT scheme, as in all other PPM schemes, routers mark (overwrite) the 16 bit IP Identification (IP ID) field of the IPv4 header of a small percentage of the packets that they forward. A FIT router marks a forwarded packet with a certain probability, q, which is a global constant among all FIT enabled routers (set to 0.04 in our experiments[4]). A packet mark is divided into three fields, as shown in Figure 2. The first field, denoted as b, is the 1-bit distance field. The second and third fields involve the router's hash.

Each FIT router pre-calculates a hash of its IP address and splits the hash into n fragments of b_{frag}-bits each, where n is a global constant. The size of each fragment, b_{frag}, is set as $15 - b_{fnum}$.[5] When marking a packet, a router randomly selects a fragment number to mark into the frag# field, and marks the corresponding fragment's bits into the hash fragment field.

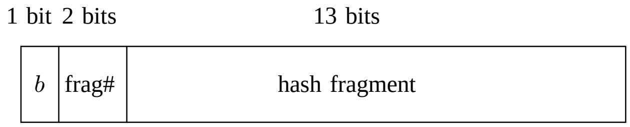

Fig. 2. FIT marking field diagram. The distance field b is one bit. In this example, the fragment number field is two bits ($b_{fnum} = 2$ bits) allowing four distinct fragments, and the remaining 13 bits are used for the hash fragment ($b_{frag} = 13$ bits).

Unlike other PPM schemes, FIT has a deterministic marking aspect. For each packet that a particular FIT router has not probabilistically marked, that same router calculates a *marking predicate* based on the packet's TTL field and distance bit. The *marking predicate* contains a calculation of the minimum bound on the distance, in FIT-enabled and legacy router hops, since the packet was last marked. If the packet was not marked for the past 32 hops then the *marking predicate* evaluates to $true$ and the packet is automatically marked by the forwarding router. The *marking predicate* is evaluated as: $(b|c - TTL_{[5..0]}) \mod 64 > 32$, where $b|c$ denotes the concatenation of the distance bit b in the packet with the global constant c, and $TTL_{[5..0]}$ denotes the six least significant bits of the TTL field. We discuss the rationale for this in more detail in Section III-C.

When marking a packet a router randomly selects a fragment number to write into the $frag\#$ field and marks the hash fragment field with the corresponding hash fragment's bits. The router also sets the 5 least-significant bits of the packet's TTL to a global constant c, and stores the 6th bit of the TTL in the distance field b. This last step allows the next FIT-enabled router, or the packet receiver, to determine the distance since the router's mark. We explain the details of calculating the distance as well as the ramifications of modifying a packet's TTL in-flight in the following section. Finally, if a router does not mark the packet then it will not change any part of the IP ID field.

```
*FIT packet marking algorithm:*
  FOR each packet $P$
    $r \overset{R}{\leftarrow} [0,1)$
    IF ($r \leq q$)
    OR (P.*dist_bit*$|c - TTL_{[5..0]} \mod 64) > 32$ THEN
      $\alpha \overset{R}{\leftarrow} [0,n)$
      P.*frag_num* $\leftarrow \alpha$
      P.*fragment* $\leftarrow H(IP)_{[(\alpha+1)\cdot b_{fnum}-1..\alpha\cdot b_{fnum}]}$
      P.*dist_bit* $\leftarrow TTL_{[5]}$
      $TTL_{[4..0]} \leftarrow c$
    ELSE
      $TTL \leftarrow TTL - 1$
```

Fig. 3. The FIT Marking Algorithm. $r \overset{R}{\leftarrow} [0,1)$ means that we select a number from the interval $[0,1)$ uniformly at random. The notation $TTL_{[5]}$ selects bit 5 of the TTL (the LSB is $TTL_{[0]}$), and $TTL_{[5..0]}$ selects the six least significant bits.

C. Calculating Distance Using a Single Bit Field

In this section, we show how FIT routers use a single bit along with TTL modification to pass marking distance information to the packet receiver. We also perform an analysis to show that our scheme preserves existing TTL semantics.

1) Distance Field in IP Traceback: In most IP traceback schemes, the distance from a marking router is kept as a 5 bit incrementing counter,[6] set to zero by the marking router and incremented by every traceback-enabled router which forwards, but does not itself mark, the packet [5], [15], [17]. Although this zero/increment scheme prevents pollution attacks, where the attacker

[4] The marking probability q is chosen as $\frac{1}{d}$ for optimizing the probability of receiving markings from routers d hops away from the packet receiver [15]. $q = 0.04$ is optimal for markings from routers at a distance of 25 hops from the reconstructing endhost.

[5] The minimum required length of the hash is $2^{b_{fnum}}(15 - b_{fnum})$ bits. Hash functions with shorter outputs can be used by concatenating a counter value to the input, evaluating the hash function once for each counter value, and concatenating the outputs (similar to the SSL/TLS protocol [6].)

[6] All mechanisms (except FIT) use saturating addition on the distance field, such that the field never overflows to show a zero distance.

sends false markings designed to misdirect the traceback, the zero/increment mechanism also has some drawbacks. First, the mechanism requires 5 bits of space to be able to identify distances commonly seen in Internet paths. Second, distance is counted as *traceback enabled* router hops since only traceback enabled routers modify the distance field. This causes an increase in false positives when large numbers of legacy routers are present. The FIT distance mechanism maintains all the properties of the zero/increment distance scheme while addressing these two issues.

2) Calculating Distance in FIT : Recall the distance-related operations a marking router performs in FIT: it sets the 5 least-significant bits of the packet's TTL field to a global constant c, and stores the sixth bit of the TTL in the distance field b. When a packet arrives at its destination, the distance at which the packet was marked is computed as: $d = (b|c - TTL_{[5..0]})\ mod\ 64$, where $b|c$ denotes concatenation of the one bit distance field b with the five bit TTL replacement constant c. Because legacy routers decrement the TTL, the FIT distance is representative of the *exact* number of hops from a marking router, rather than just the number of hops of traceback enabled routers.

3) Deterministic Marking Predicate: The reader will note that the distance calculation presented in the previous section has a range of 64 hops, roughly twice the number of hops that appear in Internet paths. However, there is a subtle attack on the FIT which violates the pollution attack prevention property, which reduces this range. Because the attacker can control the initial contents of the IP ID field, it can control the initial value of the distance field b. By selecting an initial TTL which will decrement ($mod\ 64$) past the value $b|c$ while on the path to the destination, the attacker can cause the distance calculation to wrap from 63 to 0 and to start counting up again. The countermeasure to this attack is to invalidate a portion of the distance space, and have FIT routers automatically mark any packet whose calculated distance falls within that space. For FIT, we consider the distance range [33,63] to be invalid. The marking predicate we describe in Section III-B checks whether the distance is within the invalid range, and causes the router to mark the packet if the predicate is true. Thus, a packet would have to travel at least 32 hops without encountering a FIT-enabled router before the distance calculation can wrap to zero.[7] Automatically marking packets with invalid distance values has the added benefit of increasing the number of packets that are marked. Invalidating the distance range [33,63] restricts the maximum possible distance traceable to be 32 hops away from the victim, equal to the capability of a 5-bit distance field.

4) Preservation of TTL Semantics: The FIT distance calculation relies on TTL modification of packets in-flight. It is critical, however, that FIT modification preserve existing TTL semantics. We can minimize the effect on TTL by choosing an appropriate value for the TTL replacement constant c in our marking scheme. The primary function of TTL is to cause packets in routing loops to be dropped (when their TTL reaches zero). However, we must also ensure that packets with default TTL values (such as 32, 48, 64, 128 and 256 [9]) are not dropped prematurely on path lengths likely to appear in the Internet. There is a tradeoff between low values of c, which favor the former property, and high values of c, which favor the latter property.

Using simulation, we find that the TTL replacement constant $c = 22$ preserves both of the desired properties of TTL. By modeling routing loops as long paths, we show in Table II that virtually all packets are dropped (their TTLs reach zero) after 512 hops, regardless of their initial TTL. Table III shows that a very small percentage of packets with common TTLs are dropped, and only in the infrequent cases of path lengths greater than 24.[8]

	$d = 128$	$d = 256$	$d = 384$	$d = 512$
TTL = 32	95.12%	99.97%	100%	100%
TTL = 64	70.85%	99.57%	100%	100%
TTL = 128	1.60%	83.99%	99.61%	100%
TTL = 255	0.00%	0.56%	62.58%	97.39%

TABLE II
DECAY TO ZERO ANALYSIS: PERCENT OF PACKETS DROPPED BY PATH DISTANCE AND INITIAL TTL

	TTL = 32	TTL = 48	TTL = 64
d = 16	0.00%	0.00%	0.00%
d = 24	0.82%	0.00%	0.00%
d = 32	7.10%	0.00%	0.00%
d = 48	—	4.41%	0.00%
d = 64	—	—	5.18%

TABLE III
PACKET LOSS ANALYSIS: PERCENT OF PACKETS DROPPED BY INITIAL TTL AND PATH DISTANCE

5) Marking Predicate and Percentage of Marked Packets: As mentioned previously, invalidating the distance space [33,63] and having FIT-enabled routers automatically mark packets with distances in the invalid range will increase the percentage of marked packets relative to other traceback schemes using the same marking probability, q. In fact, there are certain initial TTL values that will cause a packet to be marked 100% of the

[7] This is only likely to happen on paths with no FIT-enabled routers to begin with, in which case FIT is as vulnerable to pollution attacks as any other traceback mechanism.

[8] Note that because FIT marks probabilistically, a packet drop due to expired TTL can be solved by retransmission. Furthermore, the sender may receive an ICMP error packet and increase its initialized TTL.

time. This occurs when the TTL is in the invalid range $(b|c{-}TTL_{[5..0]})mod64 > 32$ (thus triggering the marking predicate), or will become invalid at some point along the path, regardless of whether b = 0 or 1. Figure 4 shows the probability that a packet remains unmarked given the initial TTL, for our chosen TTL replacement constant $c = 22$. The figure depicts the intuitive result that nearly half of the TTLs on a path of length 15 result in unconditionally marked packets; with longer paths having more and shorter paths having fewer such TTLs.

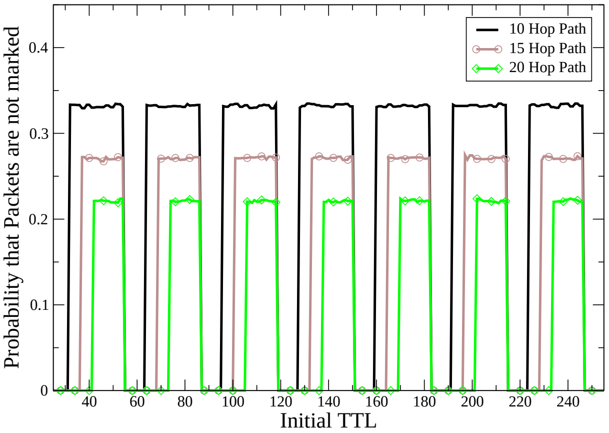

Fig. 4. Probability that packet is unmarked given an initial TTL value.

D. Map Reconstruction

FIT needs the map of upstream routers for traceback. In this section, we describe how the victim can generate this upstream router map.

From Section III-B, every packet mark consists of a IP address hash fragment, a fragment number, and a distance bit. FIT map reconstruction leverages the fact that an endhost can group together packets that traverse the same path during a TCP connection. When receiving packet markings from the same distance and TCP connection, an endhost can assume that the markings come from the same router. Thus, the endhost collects n_{map} unique fragments from a particular distance, scans through the space of all possible IP addresses, and adds the IP address whose hash matches the n_{map} fragments to the upstream router map.[9]

Map Reconstruction Accuracy Two performance metrics in Map Reconstruction require analysis: the expected number of false positives, and the number of packets required to reconstruct the IP addresses of all routers. First, we consider the number of false positives that the map reconstruction algorithm will produce. A false positive will occur when two IP addresses share a common subset of hash fragments which are received by the reconstructing endhost. The endhost will not be able to differentiate between the two IPs and will thus add both of them to the reconstructed map. If we have n distinct fragments and we need at least n_{map} fragments to reconstruct the IP address, the expected number of false positive routers reconstructed is:

$$f_p = \frac{2^{32}}{2^{(n_{map} \cdot b_{frag})}}$$

The expected number of false positive IP addresses per router to be reconstructed f_p, is independent of the number of IPs in the reconstructed map. Table IV lists f_p for candidate values of n and n_{map}.

n / n_{map} f_p	4/3 2^{-7}	8/3 2^{-4}	8/4 2^{-16}	16/4 2^{-12}	16/5 2^{-23}

TABLE IV
EXPECTED NUMBER OF FALSE POSITIVES PER ROUTER IN FIT MAP RECONSTRUCTION, FOR CANDIDATE VALUES OF n AND n_{map}.

The second performance metric for map reconstruction is the number of packets that must be sent to enable an endhost to reconstruct the IP addresses of the routers on a single path. This number provides an upper bound on the number of packets needed to reconstruct a map containing multiple paths. To quantify this, we define two probabilities: $P_{path}[k, x]$, the probability of an endhost reconstructing the IP addresses of k FIT-enabled routers on a path after receiving x packets; and $P_{ip}[i, x]$, the probability of an endhost reconstructing the IP address of the router i hops away from it after receiving x packets. Assuming that reconstructions of IP addresses of FIT routers are independent,[10] we can estimate $P_{path}[k, x]$ as $\prod_{i=1}^{k} P_{ip}[i, x]$.

To reconstruct an IP address, an endhost must receive n_{map} distinct hash fragments from the router with that IP address (as discussed above, n_{map} is selected to minimize the number of false positive routers). The probability of receiving j distinct hash fragments from a set of k total fragments after receiving y randomly selected fragments is [7]:

$$P_f[j, k, y] = \binom{k}{k-j} \sum_{v=0}^{k} (-1)^v \binom{j}{v} \left(1 - \frac{k-j+v}{k}\right)^j$$

To receive a fragment from a router at distance i, that router must mark a packet and all subsequent routers must not mark that packet. Thus, the probability of receiving

[9] Map reconstruction can be performed offline and in parallel such that only one pass over the IP address space is necessary to reconstruct all routers for which the endhost has stored n_{map} markings. A modern workstation can calculate the SHA-1 hash of all 2^{32} IP addresses in approximately half an hour.

[10] Since packets can carry only a single marking, the reconstruction probabilities are clearly not independent, but assuming independence gives us a pessimistic estimate on the number of packets required.

a packet with a fragment from a router at distance i hops from the victim, given marking probability q, is:

$$P_m[i,q] = q \cdot (1-q)^{i-1}$$

Thus, of x packets sent along a path, $xP_m[i,q]$ of them will have fragments from a router at distance i. We can now express the probability of reconstructing a router at distance i after receiving x packets ($P_r[i,x]$) in terms of P_f and P_m:

$$P_{ip}[i,x] = \sum_{v=n_{map}}^{n} P_f[v,n,x \cdot P_m[i,q]]$$

n and n_{map} are determined according to the desired false positive rate, and q is set to the inverse of the distance of the furthest router we want to reconstruct [15]. Table V shows the number of packets required to set P_{path} to 50% and 95% using candidate values of n and n_{map} for varying path lengths.

	d=5	d=15	d=25
$n=4$, $n_{map}=3$	163/265	266/414	400/623
$n=8$, $n_{map}=3$	118/175	181/266	272/400
$n=8$, $n_{map}=4$	177/255	275/392	413/590
$n=16$, $n_{map}=4$	145/190	230/300	335/442
$n=16$, $n_{map}=5$	190/250	289/382	435/574

TABLE V

NUMBER OF PACKETS TO RECONSTRUCT CERTAIN PATH LENGTHS FOR VARYING n AND n_{map} VALUES. EACH PAIR IN THE TABLE DENOTES THE NUMBER OF PACKETS NEEDED FOR 50% AND 95% PROBABILITY OF RECONSTRUCTION.

E. Path Reconstruction

The purpose of an IP Traceback mechanism is to reconstruct the IP addresses of the routers on the path from the attacker to the victim. We assume that the victim has completed the map reconstruction phase that we outline in the previous section (i.e., generated the map of upstream routers). Similar to all previous IP Traceback mechanisms, we assume that the victim has a mechanism to identify malicious packets, so that it can perform traceback.[11]

In the path reconstruction phase, the victim uses its router map and marked attack packets to reconstruct the *attack path*, which is the set of all routers that forwarded attack packets. In Section III-C we describe that the victim can detect how many routers the packet traversed since it was marked, using the one bit distance field b, the last six bits of the TTL, and the five bit TTL replacement constant c: $d = (b|c - TTL_{[5..0]})\, mod\, 64$. The fragment identifier is used to identify which subset of the hash was marked in a particular packet.

Based on these values, the victim can identify candidate attack path routers *after receiving only a single marked packet* as follows. The victim compares the hash fragment it receives with the hash fragments of all routers at the distance d in its router map, and marks any router with a matching fragment. If we have r_d routers at distance d, the b_{frag} bit hash fragment will match the marking router, as well as $r_d/2^{b_{frag}}$ false positive routers. More concretely, if we instantiate FIT with four distinct fragments ($b_{fnum} = 2$), we have 13 bits for the hash fragment ($b_{frag} = 13$); a marking from distance 8 in our map will match approximately $r_d/2^{13} = 10,000/2^{13} = 1.2$ false positive routers (Figure 6 shows the number of unique routers vs. distance from data gathered by the skitter project.). In the case that the victim's map contains a unique path from the reconstructed router to the victim, the victim can knows that the router, and all its downstream routers, are on the attack path as well.

FIT provides a significant benefit over AMS due to the fact that AMS performs link marking (i.e., each marking composed of the XOR of the hash fragments of two adjacent routers) requiring incremental (by distance) path reconstruction. In contrast, FIT path reconstruction can identify a router far away from the victim before identifying all the routers downstream from it. In some cases, FIT can even perform a rough single packet traceback. Consider the case where an attacker 15 hops away from the victim sends a single malicious packet. With probability q, the router at distance 14 (assuming that it is a FIT-enabled router), will mark the packet; and the victim will receive that marking with probability $(1-q)^{13}$. In our Internet map, we have approximately 2^{15} routers at distance 14, and in case we use a $b_{fnum} = 2$ bit field, the hash fragment size is $b_{frag} = 13$ bits, we will certainly reconstruct the correct router at distance 14, along with $2^{15}/2^{13} = 4$ false positive routers.

We now analyze the use of multiple fragments to lower the false positive rate in both single and multiple attacker traceback. In the Internet map we use in our experiments, we have about 40,000 routers at distances 11 and 12. Assuming that our hash fragment field is 13 bits long ($b_{frag} = 13$), we will still receive $40,000/2^{13} = 4.9$ false positive routers per marking, and for $b_{frag} = 12$, we will receive $40,000/2^{12} = 9.8$ false positives. To lower the false positive rate, we can require multiple markings per router. We denote the number of distinct fragments needed to reconstruct an IP address of a router as n_{path}. Requiring multiple fragments drastically reduces the number of false positives, in the case of a single attacker we have $r_d/(2^{b_{frag} \cdot n_{path}})$ false positives. For $b_{frag} = 13$ and $n_{path} = 2$, the number of false

[11] Since malicious packets contain a spoofed source IP address, the victim can detect malicious packets using a variety of techniques, e.g., TCP SYN ACK messages sent by the victim that remain unanswered, are followed by a TCP RST or ICMP Destination Unreachable packet when answered. Other indicators for malicious packets are IP source addresses that are in unallocated address blocks, contain private addresses [14], or contain a multicast source address for a multicast group that the victim did not sign up for.

positive routers at distance 11 is $40,000/2^{26} = 6 \cdot 10^{-4}$.

However, the number of false positives increases if we have multiple attackers. If we have r_{da} routers on the attack path at distance d, the false positive markings for each router on the attack path will reinforce each other. We now compute the number of false positive routers at distance d, assuming that the victim received all fragments from all r_{da} routers that forward attack traffic at distance d.[12] The probability that a specific fragment of a router not on the attack matches that fragment of a router on the attack path is:

$$p = 1 - \left(1 - \frac{1}{2^{b_{frag}}}\right)^{r_{da}}$$

Since we require at least n_{path} markings per router to add it to the attack path, the probability that a router will be a false positive is

$$p_f = \sum_{j=n_{path}}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

The number of false positive routers at distance d then is $p_f(r_d - r_{da})$.

IV. Experimental Evaluation

In this section, we complement the mathematical analysis of FIT from Section III with experimental results using representative Internet topologies, such as those provided by CAIDA's Skitter map [4]. Our experiments are divided into two sections: map reconstruction and path reconstruction.

A. Map Reconstruction

The map reconstruction experiment is as follows: every host in the entire Skitter map sends x packets to the victim (the f-root Skitter map we use has 174409 hosts). Endhosts randomize both the initial TTL and IP ID field of each of their packets. As we discuss in Section III-D, during map reconstruction the reconstructing endhost is capable of grouping together packets from the same sender by groupign packets by TCP connection.

We are interested in the accuracy (i.e., false positives vs. false negatives) as well as the speed (in terms of number of packets required from each host) in which reconstruction of the IP addresses of the routers upstream from the victim can occur. Because we are dealing with map– rather than attack path–reconstruction, a router is counted as a false positive if it is added to the reconstructed map but is not actually on any of the paths leading to the victim. Likewise, all routers not reconstructed by the victim but present on the paths in the topology are counted as false negatives.

We assume that the victim has no knowledge of the Internet topology. This means that the victim will not combine fragments collected from different paths in order to reconstruct routers common to both paths. However, false negatives are computed per-distance, so a reconstructed router in one path will count in all the paths in which it appears at the same distance from the victim.

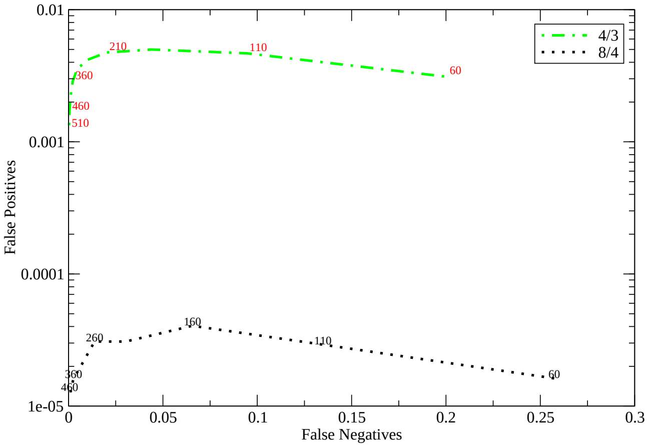

Fig. 5. Map reconstruction performance. Annotations are packets per attacker.

Figure 5 shows the Receiver Operating Curve (ROC) for two promising schemes from Section III-D,[13] the 4/3 scheme and the 8/4 scheme (the 16/5 scheme was eliminated due to poor performance in path reconstruction). The x-axis represents the rate of false negatives, and the y-axis (in logarithmic scale) represents the rate of false positives. The false positive and negative rates are computed as a ratio of the number of occurances of a false positive or negative versus the number of routers in the upstream paths. The curves for the two schemes are created by varying the number of packets sent by each endhost. Accuracy is measured as distance from the origin and speed is measured by the values of packets per endhost near each curve.

The results in Figure 5 are very strong. After as few as 200 packets per path, a victim can already reconstruct the IP addresses of over 95% of the routers in the topology. This result is already scaled to a large number of paths (174409), and it is likely that larger numbers of paths will increase the performance due to increased router overlap between paths. Finally, as predicted in the mathematical analysis of Section III-D, the 8/4 scheme performs an order of magnitude better in false positives than the 4/3 scheme due to the greater number of hash bits available to it. However, both schemes perform very well in this regard, with neither scheme producing more than 0.3%

[12] Assuming that all fragments of a router on the attack path are received is quite pessimistic in this analysis, in practice we expect that a victim would receive a smaller number of fragments from each router which would result in a smaller number of false positives.

[13] Schemes are expressed as n/n_{map}; the total number of unique fragments, n, out of which the victim must collect n_{map} fragments to reconstruct the IP address.

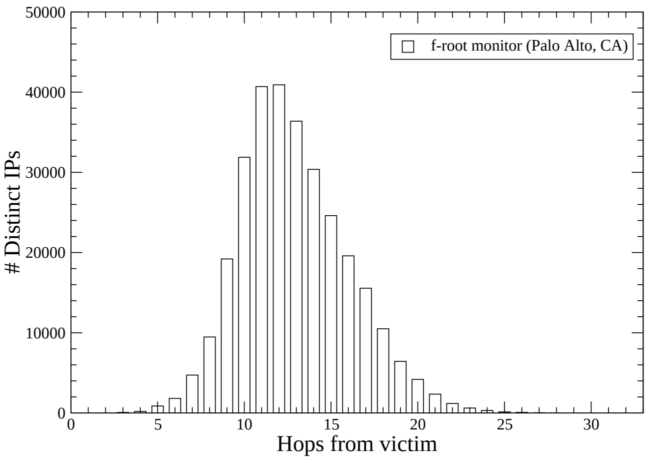

Fig. 6. Number of unique IP addresses at each hop away from the f-root Skitter monitor

false positives, regardless of the number of packets sent per endhost.

B. Path Reconstruction

Path reconstruction is the most critical performance aspect of a traceback scheme. As in map reconstruction, performance is measured in terms of false negatives, false positives, and the number of packets required from each sender. However, in path reconstruction, a host cannot correlate separate fragments since an attacker is assumed to be spoofing the IP address of each attack packet, thus preventing the victim from grouping packets together. The result is an increase in the number of false positives, as fragments from separate routers are incorrectly combined together to implicate a third router. In examining the IP address distribution of the Skitter map in Figure 6, we see that even a 13-bit hash fragment (the largest size for b_{hash} that will still allow for map reconstruction) from distance 10 will be shared by an average of approximately 5 routers. The solution is to require that multiple fragments match before a router is added to the attack path. However this allows for the possibility that two unrelated fragments cause a false positive because of the victim's inability to group them correctly.

The path reconstruction experiment is geared to show the effect of varying the number of fragments required to add a router to the attack path (changing n_{path}). We also evaluate how FIT scales with increasing numbers of attackers.

The path reconstruction experiments are similar to map reconstruction in that a given number of attackers all send x packets to the tracing endhost. False negatives, as in map reconstruction, are routers that are present on one or more of the attack paths, but are not reconstructed on any of them. False positives are routers that are present in the map but are mistakenly added to the attack path. For each path reconstruction experiment, we assume that the tracing endhost has a complete map of the upstream router tree, with no false positives. From Section IV-A we see that this is reasonable, since the false positives and false negatives are very low after the tracing endhost receives many packets.

We choose three sizes of attacker populations to show how FIT scales. The attacker populations are 100, 1000, and 5000 respectively.[14] In all of our experiments, we show the results for three candidate marking schemes (expressed as n/n_{path}): 4/3, 4/4 and 8/5. It is important to note that the difference between the 4/3 and 4/4 schemes is only in the way the tracing endhost interprets packet markings; however the difference between the 4/x and 8/5 schemes is in the way routers mark the packets.

Figure 7 shows the small attack scenario. In this graph we only show the false negative rate because no false positives were generated. In this experiment, the 4/3 scheme outperforms the 4/4 and 8/5 scheme (i.e., it provides a lower false negative rate with a smaller number of packets) largely due to the limited attacker population. With few attackers, there are few routers on the attack paths, and hence, fewer fragments to be received by the tracing endhost in total. Since each fragment will collide between multiple router IP addresses, fewer fragments means fewer false positives.

Figure 8 shows the effect of an increased attacker population on the false positive rate. Although the 4/3 scheme maintains its better false negative rate (at 305 packets it has roughly half the false negatives of the 4/4 scheme and an eighth of the 8/5 scheme), it suffers greatly in false positives. The 8/5 scheme also suffers due to the explosion of available fragments (making it easier to combine disparate fragments to falsely implicate a router), and smaller individual fragment sizes (causing more routers to be implicated per fragment). Unfortunately, attackers can use this behavior to their advantage by sending more packets and driving the curve towards higher false positives. However, the curves suggest that there is a diminishing return from such a strategy.

Finally, Figure 9 illustrates both previous points relating the number of marking fragments and their size to increased false positives. However, we see that in terms of false negatives, the 4/3 scheme still outperforms the 4/4 scheme at similar packet levels, even though its false positive rate is much higher. This result indicates that a tracing endhost can make a tradeoff between quick yet less accurate traceback (4/x where x is 1 or 2), or slower yet more accurate traceback (4/x where x is 3 or 4).

[14] It is important to note that current attacks can involve up to hundreds of thousands of attackers. However, these attack measurements include hosts from the same subnet; which is irrelevant to traceback. Furthermore, the reconstruction algorithms can be improved, as we discuss in Section V-B.

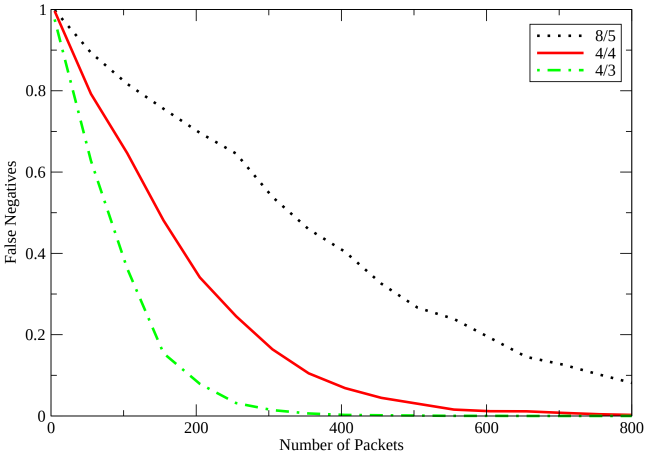

Fig. 7. Path reconstruction, 100 attackers. No false positives present.

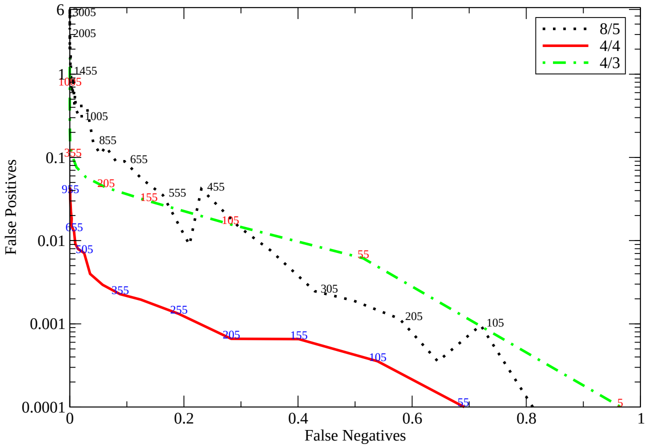

Fig. 8. Path reconstruction, 1000 attackers. Annotations are packets sent by each attacker.

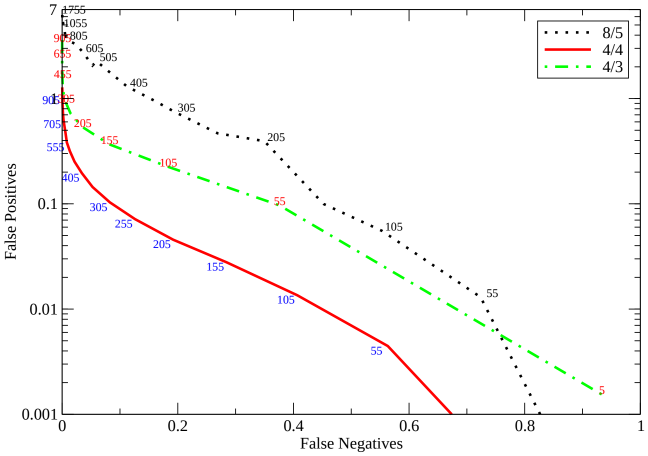

Fig. 9. Path reconstruction, 5000 attackers. Annotations are packets sent by each attacker.

V. Discussion

A. FIT Advantages

FIT offers numerous advantages over previous IP traceback proposals, and we believe that its properties make it one of the first viable traceback mechanisms.

Most of the advantages originate from the one bit distance field and our new approach for map reconstruction. First of all, the one bit distance field frees four additional bits in the IP ID field, which we can use for a larger hash fragment, which in turn greatly decreases the number of false positives. Moreover, the one bit distance field gives FIT the ability to detect legacy routers, which decrement the TTL and can thus be detected. This offers a significant advantage over AMS, which suffers from a combinatorial explosion of router choices in legacy environments (as we discuss in Section II).

FIT also uses node marking instead of edge marking. In node marking, a packet marking contains information about a single router, whereas in edge marking, a marking contains information about two consecutive traceback routers. Most previous traceback mechanisms use edge marking [5], [15], [17]. To trace back, edge marking schemes need to progressively reconstruct edge after edge starting at the victim, whereas node marking schemes can reconstruct routers at any distance as their packets arrive at the tracing host. In many cases, a single marking from a router close enough to the attacker may be sufficient to eliminate all paths except the attack path from the router map. FIT can trace back 10 or more hops on a path of length 15 after receiving only 14.3 packets, on average. The equivalent number of packets for an edge marking scheme would be on the order of thousands of packets.

Interestingly, edge marking traceback mechanisms cannot use our one bit distance field technique. In edge marking, a router needs to detect when the previous traceback router marked the packet, so that it can add its own marking to the packet. Since legacy routers decrement the TTL, and thus increment the distance, a router can not determine that it is the first router after the last marking router and thus, whether it needs to add its own marking to the packet. Another bit would be necessary for that purpose, resulting in a minimum of a two bit distance field.

B. Advanced Reconstruction Algorithms

In this paper, we present a base line path reconstruction algorithm. However, more sophisticated path reconstruction algorithms are possible. First, the path reconstruction algorithm could take advantage of the frequency of received fragments to rule out false positives. For example, consider two routers R_1 and R_2 on the attack path at a certain distance, where R_1 forwards attack traffic at a rate of α and R_2 forwards attack traffic at a rate of 10α. If we need two matching fragments to determine that a router is on the attack path, we could rule out a false positive if fragment f_1 from R_1 matches router R_3 and fragment f_2 from R_2 also matches R_3. Since fragments f_2 will appear with 10 times higher frequency than fragment f_1, we could detect that router R_3 is a false positive.

A victim could use fragment frequency information to rule out false positives even further. For example, if we reconstruct router R_3 as a false positive as just described, we could detect that it is a false positive if we do not see any fragments from its downstream router R_4. Hence, a sophisticated path analysis could further reduce our false positive rate.

C. *Traceroute*

FIT preserves the most critical TTL functionality of dropping packets in a routing loop. However, tools such as traceroute, which rely on deterministic decrementing of the TTL between routers will no longer work correctly. Legacy traceroute implementations are likely to terminate early due to packet TTLs being increased by automatic marking (a packet with a TTL of 1 will be considered to have a distance of 53 unless the distance field, d, is zero). A FIT-aware version of traceroute could provide the same functionality at the cost of an increased number of packets per trace. The details of the implementation are omitted due to space constraints.

VI. Conclusion

With the recent rise of e-crime, law enforcement and attack victims reiterate the need for a viable IP traceback mechanism. Unfortunately, current proposals for traceback mechanisms suffer from various drawbacks, including high process and storage costs, little scalability to high attacker populations and poor performance in the presence of legacy routers.

PPM schemes are particularly promising and achieved some of these properties, but they require on the order of thousands of packets from each attacker for traceback. We demonstrate a new approach, FIT, to improve packet-marking traceback. Our Fast Internet Traceback (FIT) protocol preserves the advantages of packet-marking traceback approaches and can perform traceback even after a very small number of attack packets with minimal processing overhead and without contacting any external entities. In addition, FIT handles legacy routers better than any previous mechanism, as a victim can even detect the presence of legacy routers on the attack path. In the optimal case, FIT can reconstruct a path even after a single attack packet.

FIT achieves these properties through a new approach for upstream router map reconstruction, a one-bit field to measure up to 32 hops to the distance to the marking router, node-based marking instead of edge-based marking, and a fast mechanism to identify the marking router. These techniques give FIT a previously unachieved set of properties, making it one of the only viable approaches for IP traceback.

VII. Acknowledgments

We are deeply indebted to Hiroshi Shimizu, Masami Hisada, Hiroshige Nakatani, and Toshifumi Kai from Matsushita Electric Works for their support and collaboration on this project. We would especially like to thank Hiroshige Nakatani for suggesting node marking instead of link marking in AMS. We would also like to thank the anonymous reviewers for their insightful comments.

References

[1] Micah Adler. Tradeoffs in probabilistic packet marking for IP traceback. In *Proceedings of 34th ACM Symposium on Theory of Computing (STOC)*, 2002.

[2] S. Bellovin, M. Leech, and T. Taylor. The ICMP traceback message. Internet-Draft, draft-ietf-itrace-01.txt, October 2001. Work in progress, available at `ftp://ftp.ietf.org/internet-drafts/draft-ietf-itrace-01.txt`.

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

[4] Caida. Skitter map. `http://www.caida.org/tools/measurement/skitter/`, 2000.

[5] Drew Dean, Matt Franklin, and Adam Stubblefield. An algebraic approach to IP traceback. *ACM Transactions on Information and System Security*, May 2002.

[6] T. Dierks and C. Allen. The TLS protocol version 1.0. RFC 2246, January 1999.

[7] William Feller. *An Introduction to Probability Theory and Its Applications*, volume I of *Wiley Series in Probability and Mathematical Statistics*. John Wiley & Sons, New York, third edition, 1968.

[8] Michael Goodrich. Efficient packet marking for large-scale IP traceback. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 117–126, November 2001.

[9] Cheng Jin, Haining Wang, and Kang G. Shin. Hop-count filtering: An effective defense against spoofed DDoS traffic. In *Proceedings of ACM Conference on Computer and Communications Security (CCS'2003)*, October 2003.

[10] Chris Karlof, Naveen Sastry, Yaping Li, Adrian Perrig, and J. D. Tygar. Distillation codes and applications to dos resistant multicast authentication. In *Proceedings of Network and Distributed System Security Symposium (NDSS 2004)*, February 2004.

[11] Heejo Lee and Kihong Park. On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack. In *Proceedings IEEE Infocomm 2001*, April 2001.

[12] J. Li, M. Sung, J. Xu, and L. Li. Large-scale IP traceback in high-speed Internet: Practical techniques and theoretical foundation. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.

[13] Vern Paxson. An analysis of using reflectors for distributed denial-of-service attacks. *Computer Communication Review*, 31(3), July 2001.

[14] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address allocation for private internets. Internet Request for Comment RFC 1918, Internet Engineering Task Force, February 1996.

[15] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for IP traceback. In *Proceedings of ACM SIGCOMM 2000*, August 2000.

[16] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. Hash-based IP traceback. In *Proceedings of ACM SIGCOMM 2001*, pages 3–14, August 2001.

[17] Dawn Song and Adrian Perrig. Advanced and authenticated marking schemes for IP traceback. In *Proceedings IEEE Infocomm 2001*, April 2001.

[18] Avi Yaar, Adrian Perrig, and Dawn Song. SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.