Fast Overlapping Algebraic Traceback

Dung Tien Ngo, Choong Seon Hong
Department of Computer Engineering, Kyung Hee University, Korea
Email: dungnt@khu.ac.kr, cshong@khu.ac.kr

***Abstract*— In this paper, we propose a novel scheme for IP traceback called Fast Overlapping Algebraic Traceback (FOAT) which uses *overlapping fragmentation* with *Reed-Solomon Codes*. Then we show that our FOAT scheme does not only have the same properties as the existing Fast Internet Traceback (FIT) scheme but also can build a router map with no false positives to support for traceback during attacks while FIT scheme can not.**

I. Introduction

Fast Internet Traceback (FIT) [1] is a well-known Probabilistic Packet Marking (PPM) [2] for IP traceback, where intermediate nodes, on the attack path, randomly mark the 16-bit IP Identification field of traversing packets' IPv4 headers with hash fragments of their IP addresses in order for the victim to trace the source of attacks. Before marking, every FIT router uses SHA-1 hash function to pre-calculates a hash result of its 32-bit IP address then fragments it into hash fragments. FIT has some strong properties: The victim uses an upstream-router map to provide its fast (in the number of packets received) attack path reconstruction, FIT uses the same router markings for both map reconstruction before attacks and path reconstruction during attacks. In FIT scheme, the success of path reconstruction phase depends on the correctness of map reconstruction phase. The limitation of FIT scheme is that it can not build an upstream router map with no false positives in the map reconstruction. The reason for this arises from using SHA-1 hash function because there exists collisions in the space of hash results [3]. Thus, even though the endhost receives all hash fragments from a router, it does not only add such router to the upstream router map but also other routers which share the same set of hash fragments.

In this paper, we propose a novel PPM scheme called Fast Overlapping Algebraic Traceback (FOAT) that uses a novel technique: *Overlapping fragmentation* which splits 32-bit router IP address into overlapping fragments instead of non-overlapping fragmentation in previous schemes such as FIT [1] and Fragment Marking Scheme (FMS) [2]. In addition, FOAT applies Reed-Solomon (RS) Codes [4] to encode set of overlapping fragments into points on polynomials. In fact, [5] introduced an algebraic-based traceback approach (ATA) which also uses RS Codes. However, as in FMS [2] and Advanced Marking Schemes (AMS) [6], ATA uses sub-path sampling which randomly encodes information of several routers on the attack path into each packet. With different goal that wants to reduce the number of packets required for the path reconstruction during attacks, FOAT and FIT use node sampling which just encodes randomly information of one router on the attack path into each packet. For this reason, we just compare FOAT with FIT in this paper. Through mathematical analysis and simulations, we point out that FOAT does not have all strong properties of FIT but also can build an upstream router map with no false positives.

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the C-ITRC(Convergence Information Technology Research Center) support program (NIPA-2013-H0301-13-3007) supervised by the NIPA(National IT Industry Promotion Agency), Dr. CS Hong is the corresponding author.

II. Fast Overlapping Algebraic Traceback: FOAT

A. Analysis of Design

1) Non-overlapping Fragmentation: To avoid the limitation of FIT scheme (section I), we need to find another approach so that the endhost can identify *uniquely* the router IP address based on a set of packets sent from that router. A simple solution is to split each 32-bit router IP address into *original non-overlapping fragments*. Then, as fragment-based approaches such as FIT [1] and FMS [2], a mark in the 16-bit IP Idenfification field of each packet is divided into three fields: a distance field of b_{dist} bits in order for the endhost to determine the distance of the marking router, a fragment number field of b_{fnum} bits is to distinguish $2^{b_{fnum}}$ fragments of 32-bit router IP address, and a fragment field of b_{frag} bits stores the corresponding original non-overlapping fragment while this fragment field stores a hash fragment in FIT scheme. Thus,

$$b_{dist} + b_{fnum} + b_{frag} = 16 \text{ bits}.$$

In the trivial case: 32 is divisible by b_{frag} (i.e., $32 \bmod b_{frag} = 0$), the size of each fragment of an 32-bit IP address fits the b_{frag}-bit size of fragment field. However, in the non-trivial case: 32 is indivisible by b_{frag} (i.e., $32 \bmod b_{frag} \neq 0$), there is at least one fragment with size less than b_{frag} bits. Therefore, the limitation of non-overlapping fragmentation approach is: there are wasteful bits per packet mark in the fragment field.

2) Overlapping Fragmentation: In order to solve the problem of non-overlapping fragmentation approach, we propose *Overlapping fragmentation*. This novel technique splits each 32-bit router IP address into *overlapping fragments* rather than non-overlapping fragments in previous schemes such as FIT [1], FMS [2], and ATA [5]. Due to the fact that the fewer number of fragments, the fewer number of packets required

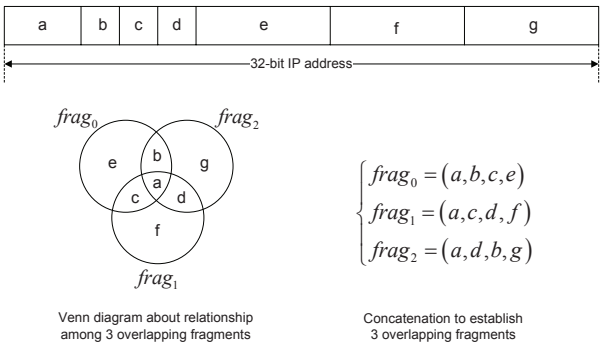

Fig. 1. Overlapping Fragmentation.

to collect, it is reasonable to split each 32-bit router IP address into minimum number of overlapping fragments which equals $\lceil 32/b_{frag} \rceil$.

In order to determine IP address of the marking router, the endhost must receive all $\lceil 32/b_{frag} \rceil$ distinct fragments. Therefore, it is neccessary for the marking router to distinguish such distinct fragments. In other words, the fragment number field of each packet mark must distinguish the forwarding fragment with other fragments of the marking router:

$$2^{b_{fnum}} \geq \lceil 32/b_{frag} \rceil \ . \tag{1}$$

For $(b_{dist}, b_{fnum}, b_{frag})$ satisfies (1), the limitation of overlapping fragmentation approach arises when the number of fragments $\lceil 32/b_{frag} \rceil$ that a router has is less than the maximum number of distinct fragments $n = 2^{b_{fnum}}$ that the fragment number field on a marked packet could distinguish. For example, $(b_{dist}, b_{fnum}, b_{frag}) = (1, 2, 13)$ then $n = 4$, or $(b_{dist}, b_{fnum}, b_{frag}) = (1, 3, 12)$ then $n = 8$, while there are only 3 overlapping fragments in such cases. It means that, the fragment number field is not exploited all its capacity of distinguishing distinct fragments.

3) Reed-Solomon Codes for Overlapping Fragments: To solve the limitation of overlapping fragmentation, we propose to apply Reed Solomon Codes [4] for overlapping fragments: instead of forwarding an original message as a sequence of $f = \lceil 32/b_{frag} \rceil$ overlapping fragments $frag_0, frag_1, \ldots, frag_{f-1}$, each router forwards the encoding message as a *longer* sequence of $n = 2^{b_{fnum}}$ distinct points $e_0, e_1, \ldots, e_{n-1}$ on the following polynomial

$$P(x) = frag_0 + frag_1 x + \ldots + frag_{f-1} x^{f-1} \tag{2}$$

over the Galois field $GF(p)$ with $p = 2^{b_{frag}}$. Therefore, our encoding $e_0, e_1, \ldots, e_{n-1}$ would consists of the values $P(0), P(1), \ldots, P(n-1)$. Notice that we need $p > n$ to have n distinct points. After receiving enough f points from n distinct point, the endhost could determine uniquely the polynomial $P(x)$ by using Lagrange interpolation. Under the view point of linear algebra, the determination of polynomial $P(x)$ from f received distinct points is to solve the following matrix equation over $GF(p)$.

$$\begin{pmatrix} 1 & x_1 & \ldots & x_1^{f-1} \\ 1 & x_2 & \ldots & x_2^{f-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_f & \ldots & x_f^{f-1} \end{pmatrix} \begin{pmatrix} frag_0 \\ frag_1 \\ \vdots \\ frag_{f-1} \end{pmatrix} = \begin{pmatrix} P(x_1) \\ P(x_2) \\ \vdots \\ P(x_f) \end{pmatrix}.$$

Because f received points are distinct, the matrix is a Vandermonde matrix which has full rank, i.e., f overlapping fragments $frag_0, frag_1, \ldots, frag_{f-1}$ are determined uniquely.

B. Packet Marking

In our FOAT scheme, a packet mark in the 16-bit IP Identification field of packet's IPv4 header is divided into three fields, as shown in Fig. 2: a distance field of $b_{dist} = 1$ bit , a point id field of b_{point_id} bits, and a point value field of $b_{point} = 15 - b_{point\_id}$ bits.

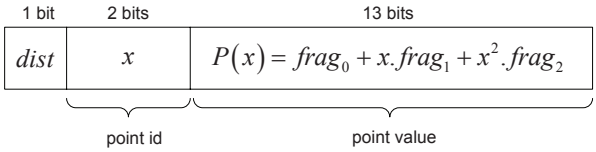

Fig. 2. FOAT marking field diagram. The distance field $dist$ of $b_{dist} = 1$ bit. In this example, the point id field of $b_{point\_id} = 2$ bits allowing $n = 4$ distinct points. The remaining $b_{point} = 13$ bits are used to store corresponding point value which is evaluated from a polynomial defined by three 13-bit overlapping fragments.

For marking, each FOAT router pre-calculates $n = 2^{b_{point\_id}}$ distint points evaluated from the polynomial in (2) which is defined by its $f = \lceil 32/b_{point} \rceil$ overlapping fragments $frag_0, frag_1, \ldots, frag_{f-1}$ over the Galois field $GF(p)$ with $p = 2^{b_{point}}$. Such pre-calculated n points correspond to n entries $e_0, \ldots, e_{n-1}$ stored in the memory of each FOAT router, where each entry e_i contains: a point id field $e_i.point\_id$, and a corresponding point value field $e_i.point\_val$ which is evaluated from (2).

Every FOAT router marks (overwrites) the 16-bit IP Identification field of traversing packet's IPv4 header with probability p. Once a router decides to mark, it will randomly pick an entry e out of n pre-calculated point entries stored in its memory: $e \xleftarrow{R} \{e_0, \ldots, e_{n-1}\}$, then writes $e_i.point\_id$ and $e_i.point\_val$ into corresponding point id field $P.point\_id$ and point value field $P.point\_val$ of the packet's marking field. After that, the router runs the 1-bit distance mechanism together with TTL modification technique as in FIT scheme [1]: sets $TTL_{[4..0]}$ (the 5 least significant bits of the packets TTL field) to a global constant $const$ (22 is the optimal value), and stores $TTL_{[5]}$ (the 6^{th} bit of the TTL field) in the distance field $dist$ in order for the next FOAT router, or the receiver, to determine the distance from the marking router. FOAT packet marking algorithm at the marking router is described in Algorithm 1.

Comparing the marking field diagram (Fig. 2) and the packet marking algorithm (Algorithm 1) in our FOAT scheme with FIT's [1, Fig. 2 and Fig. 3], it is clear that FOAT and FIT

Algorithm 1 FOAT packet marking algorithm

for each packet P **do**
  Pick u uniformly at random from $[0, 1]$
  if $(u < p)$ OR$(P.dist|const - TTL_{[5..0]} \bmod 64) > 32$ **then**
    /*decide to overwrite*/
    $P.dist \leftarrow TTL_{[5]}$
    $TTL_{[4..0]} \leftarrow const$
    $e \overset{R}{\leftarrow} \{e_0, \ldots, e_{n-1}\}$
    $P.point\_id \leftarrow e_i.point\_id$
    $P.point\_val \leftarrow e_i.point\_val$
  else
    /*decide not to overwrite*/
    $TTL \leftarrow TTL - 1$
  end if
end for

are similar except the content of marking onto each packet: 2-bit point id field compared with 2-bit fragment number field, 13-bit point value field compared with 13-bit hash fragment field.

C. Map Reconstruction

In order to traceback during attacks, FOAT build an upstream router map before. As FIT scheme [1], the FOAT map reconstruction exploits the fact that an endhost can group packets that travel the same path during a TCP connection. Let us denote n/n_{map} FOAT as a FOAT scheme where every FOAT router has n distinct points and the endhost collects n_{map} distinct points from a particular distance, scans through all 2^{32} possible IP addresses, and adds the IP addresses which also has such n_{map} received distinct points to the upstream router map. Notice that $n/n_{nmap} = 4/x$ and $n/n_{nmap} = 8/x$ indicates two FOAT schemes with different marking fields on packets. Specifically, $4/x$ FOAT scheme (Fig. 2) corresponds to $(b_{dist}, b_{point\_id}, b_{point}) = (1, 2, 13)$ while $8/x$ FOAT scheme corresponds to $(b_{dist}, b_{point\_id}, b_{point}) = (1, 3, 12)$.

FOAT is more perfect than FIT because the endhost in FOAT just needs to collect $n_{map} = f$ distinct points in order to build an upstream router map with no false positives while it can not in FIT even though the endhost receives all distinct hash fragments from the marking router (section I). Whereas, regardless of $4/x$ or $8/x$ FOAT scheme, the endhost just needs to collect 3 distinct points from a particular distance within a TCP connection, it could determine exactly the marking router's IP address by solving the following matrix equation with 3 unknown overlapping fragments $frag_0, frag_1, frag_2$, without scanning through the space of all possible IP addresses to find matches:

$$\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix} \begin{pmatrix} frag_0 \\ frag_1 \\ frag_2 \end{pmatrix} = \begin{pmatrix} P(x_1) \\ P(x_2) \\ P(x_3) \end{pmatrix} .$$

D. Simulation Results

Our goal in this section is to show simulation results comparing FOAT with FIT in terms of average number of packets required for the endhost to reconstruct all router IP addresses, with the lowest number of false positives, through 1000 tests per each specific path length d from 1 to 31 hops

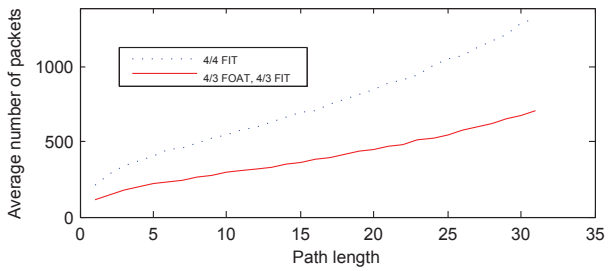

Fig. 3. Experimental results for map reconstruction by average number of packets needed to reconstruct paths of varying lengths in FOAT scheme and FIT scheme via 1000 tests per path length with marking probability $p = 1/25$.

in the map reconstruction. Every router on the path randomly marks its one of n points onto a packet with probability $p = 0.04$ that is the optimal value for PPM schemes [2]. From our simulation results in Fig. 3, $4/3$ FIT and our FOAT $4/3$ require the same average number of packets due to the fact that these two schemes are similar except the content of marking on packets. The important thing is that there are no false positives in the map reconstruction of our FOAT $4/3$, compared with $4/3$ FIT (section II-C). In addition, in order to have the lowest number of false positives in the map reconstruction, the endhost must run $4/4$ FIT in FIT scheme which does not only requires more average number of packets than our $4/3$ FOAT scheme but also still has false positives in the upstream router map (section II-C). Because the comparison between $8/x$ FIT and our $8/x$ FOAT scheme is similar, we do not show simulation results of such schemes in Fig. 3.

III. Conclusion

In this paper, we has proposed Fast Overlapping Internet Traceback (FOAT) scheme where every FOAT router randomly marks Reed-Solomon Codes of its overlapping fragments on traversing packets. Through mathematical analysis and simulations, we showed that FIT and our FOAT scheme are similar except the content of marking in the packet's 16-bit marking field. Therefore, our FOAT scheme has all strong properties of FIT scheme such as: fast attack path reconstruction and using the same router markings for both map and path reconstructions. In addition, FOAT can build a router map with no false positives to support for traceback during attacks while FIT can not.

References

[1] A. Yaar, A. Perrig, and D. Song, "FIT: Fast Internet Traceback," in *Proc. of IEEE INFOCOM 2005*, 2005.

[2] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback," *Networking, IEEE/ACM Transactions on*, 2001.

[3] X. Wang, Y. L. Yin, and H. Yu, "Finding Collisions in the Full SHA-1," in *International Crytology Conference*, 2005.

[4] M. Mitzenmacher. www.eecs.harvard.edu/ michaelm/CS222/eccnotes.pdf.

[5] D. Dean, M. Franklin, and A. Stubblefield, "An Algebraic Approach to IP Traceback," *ACM Trans. Inf. Syst. Secur.*, 2002.

[6] D. X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," in *Proc. of IEEE INFOCOM 2001*, 2001.