2014 IEEE 7th International Conference on Service-Oriented Computing and Applications

An Efficient and Adaptive IP Traceback Scheme

Kayoko Iwamoto
and Masakazu Soshi
Graduate School of Information Sciences,
Hiroshima City University
Asa-Minami-Ku, Hiroshima, 731-3194 Japan
Email: soshi@hiroshima-cu.ac.jp

Takashi Satoh
Faculty of Engineering,
The University of Kitakyushu
Hibikino 1-1, Wakamatsu-ku, Kitakyushu, 808-0135 Japan
Email: tsatoh@kitakyu-u.ac.jp

***Abstract*—IP traceback is considered to be one of the promising countermeasures against Distributed Denial of Service (DDoS) attacks. IP traceback protocols must be effective as well as simple enough to be efficiently executed. However, there is almost no such an IP traceback protocol.**

In this paper, we consider an IP traceback protocol proposed by Muthuprasanna and Manimaran [1] (STE scheme for short) and shall propose a new, efficient, and adaptive IP traceback scheme, which is partly based on STE. Simply speaking, our scheme is efficient since it adaptively changes marking probabilities to decrease the number of marking bits. In this paper, we conduct theoretical and numerical analyses of our scheme in detail and show that our scheme is more efficient than STE in terms of marking bit length and the number of packets for attack path recovery. The result is also supported by simulation experiments.

***Keywords*-security; IP traceback; network; protocol**

I. Introduction

Denial of Service (DoS) attacks have been a serious security concern to the today's Internet. In a DoS attack, an attacker tries to shut down or at least disrupt a target host machine by sending a huge amount of packets to the target, which is called a *victim* (besides we call a path along which an attack packet traverses from one attacker to the victim an *attack path*). Even worse, in recent years a *Distributed DoS attack* (DDoS attack for short), where many attackers simultaneously mount DoS attacks, has been prevailing [2]. There often exist more than one thousand attackers (or malicious bots) in DDoS attacks.

Against DoS/DDoS attacks, we consider *IP traceback* [1], [3], [4] to be an effective countermeasure. An IP traceback scheme consists of the following two phases:

1) *Marking phase*: Each router on an attack path marks packets with (partial) information about the path.
2) *Traceback phase*: When the victim receives a sufficient number of marked packets, it uses the information to reconstruct the attack path and tries to trace the attack back toward a potential attacker.

IP traceback protocols must be effective as well as simple enough to be efficiently executed. However, there is almost no such an IP traceback protocol, although traceback protocols have so far been extensively studied.

In this paper, we consider an IP traceback protocol proposed by Muthuprasanna and Manimaran [1] (STE scheme for short) and shall propose a new, efficient, and adaptive IP traceback scheme, which is partly based on STE. Simply speaking, our scheme is efficient because it adaptively changes marking probabilities of routers to decrease the number of marking bits in packets. The detail is given in section III.

Additionally, in this paper we perform theoretical analysis of our proposed scheme in detail. As a result, we show that our proposed scheme outperforms STE in terms of the number of required marking bits and the number of packets for attack path recovery.

Finally, we carry out simulation experiments and then demonstrate that in our scheme the number of packets needed for traceback is less then that of STE scheme.

This paper is organized as follows. In section II, we discuss previous related work and then propose a new IP traceback scheme in section III. We conduct theoretical and numerical analyses in detail in sections IV and V, respectively. Section VI conducts simulation experiments. Finally, we conclude this paper in section VII.

II. Related Work

In this section we discuss previous IP traceback protocols. In particular, we present an IP traceback protocol based on space-time encoding, which is proposed by Muthuprasanna and Manimaran [1], and point out the disadvantages.

A. Previous IP Traceback Protocols

In this paper, for IP traceback protocols, we consider *probabilistic packet marking* (PPM for short) protocols. In PPM protocols, each router on an attack path stores path information with some probability onto packets that it receives and then forwards them to the next router on the path [3], [4]. We discuss some of the important ones below.

Yu et al. proposed an IP traceback protocol that takes advantage of entropy variation [5]. However, their flow monitoring algorithm and IP traceback algorithm are intricate and we do not know if it would work efficiently as expected under DoS/DDoS attacks in practical network environments.

RIHT [6] is classified as a hybrid IP traceback and it is motivated by lessening storage burden on each router under DoS/DDoS attacks. Unfortunately, RIHT also has some disadvantages. For example, if a participating router on an attack path fails due to DoS/DDoS attacks, then the upstream links

978-1-4799-6833-6/14 $31.00 © 2014 IEEE
DOI 10.1109/SOCA.2014.19

stored in the router towards the attacker may be completely lost and subsequently we could not recover the attack path.

We consider an IP traceback scheme based on space-time encoding [1] to be interesting and be one of the most promising ones. In the next section we introduce the protocol.

B. IP Traceback based on Space-Time Encoding

Muthuprasanna and Manimaran proposed an IP traceback scheme based on space-time encoding [1]. Here 'space time encoding' is an efficient encoding scheme of attack trees under DDoS attacks, where an *attack tree* is composed of the victim as the root of the tree, attackers as the leaves, and intermediate routers on attack paths as the tree nodes. In the subsequent parts of this paper, with respect to a router R on an attack path, we let an *upstream router* mean a router located on the path between the attacker and R. Similarly, a *downstream router* is a router between R and the victim.

In this paper, however, we do not consider the detail of space-time encoding itself any further. Instead, we concentrate on the IP traceback scheme proposed in [1] and we call it *STE scheme* for short.

In STE scheme, a number from 0 to $n-1$ is assigned to each physical interface connecting an adjacent upstream router[1]. Hence every interface number can be represented in $\lceil \log_2 n \rceil$ bits. However, since the space in each IPv4 packet header where attack path information is to be stored is severely limited, we divide the $\lceil \log_2 n \rceil$ bits into k labels, each of which is m bits long. We require that $k \times m \geq \lceil \log_2 n \rceil$.

Now we present an outline of the packet marking algorithm of STE scheme[2] as follows. Let us suppose that a router R receives a packet P. With a probability p, R chooses a new label number c. Let ℓ be the c-th label value of the incoming interface of P. Then R marks (appends) $0\|c\|\ell$ in P. Here '$\|$' means the bit string concatenation. Otherwise (that is, with probability $1-p$), R appends $1\|\ell$ in P where ℓ is the label value of the label number that the upstream router closest to R wrote in P.

C. Disadvantages of STE Scheme

In this section, we shall point out the problems of STE scheme. First, in STE, notice that with probability p newly marked information is $(1 + \lceil \log_2 k \rceil + m)$ bits long, otherwise $(1+m)$ bits long. Therefore, for a packet, *the smaller the probability* p *is, the more efficient STE becomes in terms of the number of marking bits*. Thus in the extreme case of $p = 0$, STE is the most efficient. This case implies that only the first router on an attack path chooses a label number c and stores $0\|c\|\ell$ on a packet P[3]. If each subsequent router on the attack path receives P, then it gets c from P and stores (appends) $1\|\ell$ on P.

However, the extreme case above has the obvious drawback. Namely, the drawback is that it is substantially difficult to find the first router (other than the attacker) on the attack path. Consequently the attacker can easily pretend to be the first router because he can forge packets in any way. Thus the attacker might be able to mount any attack against STE in the extreme case.

To sum up, we can now see that although a smaller probability p is desirable in STE, it must not be zero (i.e., the extreme case is not allowable). However, unfortunately, it is not known how to determine an appropriate value of p.

Next, we discuss another problem of STE scheme. As demonstrated above, in STE each router on an attack path marks information on packets independently with probability p. In other words, STE does not adaptively change the behavior to take advantage of valuable marking information stored on packets. Therefore, in STE, there is still room for improvement of efficiency if every participating router can exploit the valuable information.

III. Our Proposal

In this section, we shall propose a new, efficient, and adaptive IP traceback scheme, which is partly based on STE. Note that in the subsequent sections, we focus on DoS attacks only, and are not much concerned with DDoS attacks. However, 'space-time encoding' is originally devised for efficient encoding of attack trees in DDoS attacks [1]. Our proposed scheme extends and improves STE and thus it should not be difficult to apply our scheme to DDoS attack situations.

A. Our Scheme

We now present our proposed scheme in this section. However, before that, in order to understand the basic idea, first we consider how STE scheme can be improved.

To begin with, we introduce a concept of a leader. A *leader* is a router on an attack path that appends a label number and the corresponding label value into a packet. For example, we see that in STE each router becomes a leader with probability p and a non-leader with $1-p$.

Here, notice that in STE, when a router R receives a packet P, P must have at least one leader to designate a label number so that the succeeding routers on the path can mark label values of the label number. Furthermore, as discussed in section II-C, the number of leaders for P should be small for efficiency. On the other hand, if the number of leaders for P is always one (see the extreme case demonstrated in section II-C), attackers could mount various attacks.

Now we can explain our basic idea for an efficient traceback scheme: When a router R receives a packet P, *if* R *finds that one or more leaders have already marked* P*, then the*

[1] Originally in [1], the authors supposed that interfaces are numbered *from 0 to* n, but strictly speaking, in that case interface numbers cannot be represented in $\lceil \log_2 n \rceil$ bits. Therefore, to be more precise, we assign a number from 0 to $n-1$ to each interface number in this paper.

[2] The original IP traceback in [1] has to find the first router on an attack path, which determines the label number of labels to be written later. However, the first router must be other than the attacker and it is quite difficult to find such a router in general. Thus in [1], to overcome such a difficulty, the authors suggest the probabilistic packet marking, which is presented here in this paper.

[3] Notice that in this case there must exist at least one router which specifies a label number, otherwise no routers on the attack path can determine label values to be marked.

necessity for R to become a new leader is greatly reduced for efficiency in terms of marking bit length. Instead, simply R has only to become a non leader, i.e., to store the label value corresponding to a label number that some of the upstream leaders have already marked on P.

In order for an IP traceback scheme to have a property described above, what it has to do is, roughly speaking, just to adaptively decrease marking probabilities when P has many leaders. For that purpose, we shall consider *a packet marking scheme where the marking probability is inversely proportional to the number of leaders marked on a packet.*

To be more specific, let us suppose that a router R receives a packet P and the number of (upstream) leaders presented in P is i. Then R becomes a leader with the marking probability p_i, which is defined as

$$p_i = \frac{p}{i+1}, \tag{1}$$

where p is an initial marking probability. That is to say, with the probability given above, R stores (appends) both a label number and the corresponding label value onto P. Otherwise (with probability $1 - p_i$), R becomes a non-leader and writes the label value corresponding to a label number that some upstream leader has already marked on P.

Now we are in a position to actually propose a new adaptive IP traceback scheme. Below, assume that a packet P has just reached a router R. Additionally, let us suppose that each router has a label counter c, an initial value of which is arbitrary. Let T be the marking area[4] in P and b be a *marking type bit* to distinguish two types of label information from each other. Then putting them above all together, R executes the algorithm given in Fig. 1, which is our proposed scheme. After the algorithm in Fig. 1 terminates, P is sent to the next adjacent router on the attack path.

In what follows, we call information marked in step MA-0 (i.e., $b\|c\|\ell$) *marking information of type 0*, and information marked in step MA-1 (i.e., $b\|\ell$) *marking information of type 1*. We can consider marking information in STE scheme in a similar vein.

The traceback protocol of our scheme is almost the same as in [1] and is omitted due to the space limitation.

```
Get the the number of leaders given in $P$ by scanning
  $T$ of $P$ and counting the number of marking type bits
  of zero.
$i \leftarrow$ the number of leaders
**if** $i = 0$ **or** with probability $p_i = \frac{p}{i+1}$ **then**
    /* step MA-0 (become a leader) */
    $b \leftarrow 0$ ; $c \leftarrow (c+1) \bmod k$
    $\ell \leftarrow$ the $c$-th label value of the incoming interface
            of $R$ for $P$
    marks $b\|c\|\ell$ at the beginning of $T$ of $P$
    shift $T$ to the right by $1 + \lceil \log_2 k \rceil + m$ bits
**else**
    /* step MA-1 (become a non leader) */
    $b \leftarrow 1$
    $c \leftarrow$ the label number that the upstream leader
            closest to $R$ marked in $P$
    $\ell \leftarrow$ the $c$-th label value of the interface
    mark $b\|\ell$ at the beginning of $T$
    shift $T$ to the right by $1 + m$ bits
**endif**
```

Fig. 1. Our proposed scheme

IV. Analysis

In this section we examine our proposed scheme extensively. Our analysis shows that our scheme is superior to STE scheme [1] in terms of the number of marking bits and the number of packets required for recovering an attack path.

A. Analysis of the Number of Marking Bits for Traceback

1) The Number of Marking Bits for Traceback in STE Scheme: Here we consider the number of marking bits in STE scheme necessary for recovering the attack path. Let r (≥ 1) be the number of routers on the attack path between an attacker and the victim. Note that in this case at least one router on the path must mark a packet with marking information of type 0 (i.e., marking type bit, a label number and the label value). Except such a router, the expected number of routers which store marking information of type 0 into a packet is given by $(r-1)p$. On the other hand, the expected number of routers which write marking information of type 1 (i.e., marking type bit and a label value) into a packet is $(r-1)(1-p)$. Consequently the average number of marking bits necessary for traceback is

$$(1+(r-1)p)\cdot(1+\lceil \log_2 k \rceil+m)+(r-1)(1-p)\cdot(1+m) \ . \tag{2}$$

As we imagine, Eq. (2) is a monotonically increasing function of p and hence the value is minimized when $p = 0$. However, the disadvantage of this case was discussed in detail in section II-C.

2) The Number of Marking Bits for Traceback in Our Scheme: Now we proceed to our proposed scheme. As above, r denotes the number of routers on an attack path. In addition, let us suppose that the number of leaders that appear in a packet P is i (≥ 1). Packet marking procedures for P have been done in a following manner.

(1) For simplicity, we assume that the first router other than the attacker on an attack path becomes a leader with probability 1 and marks type 0 marking information on P. Each of subsequent n_1 (≥ 0) routers does not become a leader with probability $1-p_1 = 1-\frac{p}{2}$ (see also Eq. (1)) and stores type 1 marking information.
(2) (n_1+2)-nd router on the path becomes a leader with probability $p_1 = \frac{p}{2}$ and marks type 0 marking information. Each of subsequent n_2 (≥ 0) routers does not become a leader with probability $1 - p_2 = 1 - \frac{p}{3}$ and stores type

[4] Dean et al. [3] argued that we can use 25 bits in each IPv4 packet header for IP traceback.

1 marking information, which corresponds to the label number given by the (n_1+2)-nd router.

$\vdots$

(i) $(n_1+n_2+\cdots+n_{i-1}+i)$-th router becomes a leader with probability $p_{i-1}=\frac{p}{i}$ and marks type 0 marking information. Each of subsequent n_i (≥ 0) routers does not become a leader with probability $1-p_i=1-\frac{p}{i+1}$ and stores the corresponding label value (actually, type 1 marking information) only.

We obtain the probability of the event above as

$$\begin{aligned} &1\cdot(1-p_1)^{n_1}\cdot p_1\cdot(1-p_2)^{n_2}\cdot p_2\\ &\quad\cdots(1-p_{i-1})^{n_{i-1}}\cdot p_{i-1}\cdot(1-p_i)^{n_i}, \end{aligned} \tag{3}$$

where we suppose that $n_1\geq 0, n_2\geq 0,\ldots,n_i\geq 0$ and $n_1+n_2+\cdots+n_i=r-i$.

Note that every leader stores marking information of type 0, which is $1+\lceil\log_2 k\rceil+m$ bits long. Moreover, every non-leader writes marking information (of type 1) of $m+1$ bits. In summary, the expected number of marking bits B is given by:

$$\begin{aligned} B=\sum_{i=1}^{r}\sum_{n_1+n_2+\cdots+n_i=r-i} & p_1p_2\cdots p_{i-1}(1-p_1)^{n_1}(1-p_2)^{n_2}\\ &\cdots(1-p_i)^{n_i}\times\left(i(1+\lceil\log_2 k\rceil+m)+(r-i)(1+m)\right)\ , \end{aligned} \tag{4}$$

where the second summation is computed over $n_1\geq 0, n_2\geq 0,\ldots,n_i\geq 0$ such that $n_1+n_2+\cdots+n_i=r-i$.

B. Analysis of Traceback

In this section we evaluate the number of packets for reconstructing an attack path. As in section III-A, we assume that the number of routers other than the attacker on an attack path is r.

1) Analysis Model: Now we give a model for our analysis. In both of STE and our scheme, each router becomes a leader with some probability, say, q, and a non-leader with $1-q$. Therefore, as in [1], [4], for simplicity, we assume that each router marks packets with label values in an independent and uniform manner. Thus each router stores i-th label value $(i=1,\ldots,k)$ into a packet with the same probability $\frac{1}{k}$, which does not depend on q. In other words, when a packet reaches the victim, it contains a list of label values each of which is chosen by a router independently and uniformly at random.

Under the assumption above, the number of required packets for attack path recovery can be evaluated in the same way for STE and our scheme. Therefore hereinafter we apply the same analysis in sections IV-C and V-B to both STE and our scheme. However, needless to say, such an analysis is rather conservative for our scheme. Namely, the obtained value of the number of packets is merely an upper bound of the average number of packets for traceback in our scheme and hence we expect that our scheme can work more efficiently than STE. That is actually shown in section V-B.

C. Analysis

Now, in order to evaluate the number of packets for attack path recovery, we apply a variation of the famous *coupon collector problem* [7] to our analysis. That is, suppose that there exist r coupon issuers, each of which publishes k kinds of coupons. Then a collector gets one coupon from every issuer at one time. So the total number of coupons that the collector obtains each time is r. Therefore, if we regard the collector and the issuers as the victim and the routers respectively, then the expected number of the packets necessary for path recovery is nothing but the expected number of rounds that the collector eventually has all k kinds of coupons from each of r issuers (i.e., the total number of the coupons is at least kr). Notice that the required number of rounds that the collector acquires all k kinds of coupons from all r issuers is just the maximum number of coupons that some issuers have yielded until their k kinds of coupons are collected.

Let us turn to the actual analysis. First we define some random variables as follows.

X_i: the number of coupons issued by issuer i $(i=1,\ldots,r)$ for the collector to get all k kinds of coupons of issuer i.

X_{ij}: the number of coupons required to get a coupon of j-th kind from issuer i, given that the collector has already obtained $j-1$ kinds of coupons $(i=1,\ldots,r,$ $j=1,\ldots,k)$ from issuer i.

We can establish the relationships among the random variables X_i and X_{ij} as

$$X_i=\sum_{j=1}^{k}X_{ij}, \tag{5}$$

and

$$\Pr(X_{ij}=\ell)=\left(\frac{j-1}{k}\right)^{\ell-1}\cdot\left(1-\frac{j-1}{k}\right), \tag{6}$$

where $\ell\geq 1$ and $X_{i1}=1$ for arbitrary issuer i.

Furthermore, from Eqs. (5) and (6), for $n_m\geq 1, 1\leq m\leq k$ we have

$$\begin{aligned} \Pr(X_i=\ell)&=\sum_{n_1+n_2+\cdots+n_k=\ell}\Pr(X_{i1}=n_1,\ldots,X_{ik}=n_k)\\ &=\sum_{n_1+\cdots+n_k=\ell}\left(\prod_{m=1}^{k}\left(\frac{m-1}{k}\right)^{n_m-1}\left(1-\frac{m-1}{k}\right)\right)\ , \end{aligned} \tag{7}$$

where $\ell\geq k$.

Based on the discussion above, what we have to do is to derive the expected maximum value among $X_1, X_2, \ldots$, and X_r. Under our assumption, $X_1, X_2, \ldots$, and X_r independently follow the identical probability distribution, which can be expressed by Eq. (7). So we let a single random variable X represent $X_1, X_2, \ldots$, and X_r. Furthermore, suppose that random variable $X_{(r)}$ $(\geq k)$ denotes the maximum value among $X_1, X_2, \ldots, X_r$.

Hence we achieve

$$\Pr(X_{(r)} = \ell) = \Pr(X \leq \ell)^r - \Pr(X \leq \ell - 1)^r$$
$$= \left(\sum_{m=k}^{\ell} \Pr(X = m) \right)^r - \left(\sum_{m=k}^{\ell-1} \Pr(X = m) \right)^r \quad .(8)$$

In summary, we know that the average number $E[X_{(r)}]$ of packets for path recovery is obtained by:

$$E[X_{(r)}] = \sum_{\ell=k}^{\infty} \ell \cdot \Pr(X_{(r)} = \ell) \quad . \tag{9}$$

Eq. (9) can be evaluated from Eqs. (7) and (8).

V. Numerical Analysis

Based on the analysis in section IV, in this section we conduct a numerical analysis of STE and our scheme in order to observe their performance and behavior in more practical situations.

A. The Number of Bits Required for Traceback

Here we discuss the number of marking bits necessary for recovering an attack path in STE and our scheme.

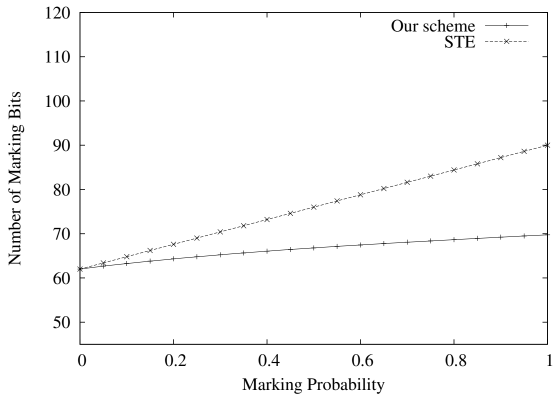

Fig. 2. The number of bits required for traceback ($k = 3, m = 3$).

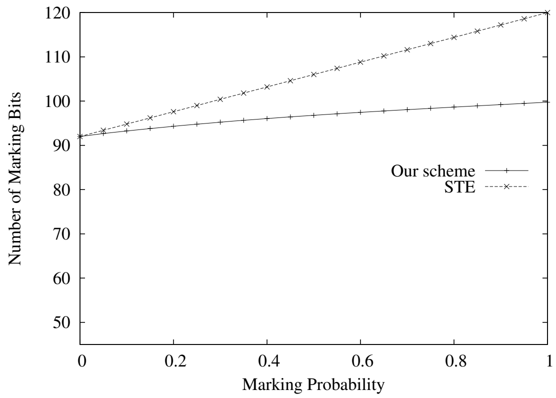

Fig. 3. The number of bits required for traceback ($k = 3, m = 5$)

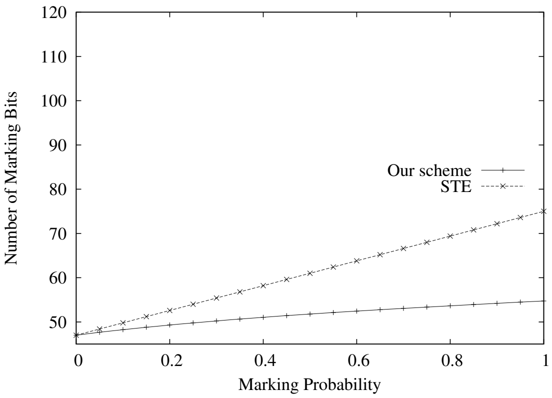

Fig. 4. The number of bits required for traceback ($k = 4, m = 2$)

We compute Eqs. (2) and (4) with the initial marking probability p from 0.0 to 1.0 (see also Eq. (1)). For r (the numbers of routers), we set $r = 15$. With respect to k (the number of labels for one interface number), and m (label bits), we consider the following settings: (i) $k = 3, m = 3$ (Fig. 2), (ii) $k = 3, m = 5$ (Fig. 3), (iii) $k = 4, m = 2$ (Fig. 4)).

From Figs. 2, 3, and 4, we can easily see that the numbers of marking bits in STE and our scheme are monotonically increasing functions of p. This is because when p is large, the probability that a router becomes a leader is also large (see Eq. (1)). In consequence, when p is large, more marking bits are required since the number of routers that write marking information of type 0 gets larger.

Moreover, from these figures, notice that the larger p becomes, the larger the difference of the numbers of marking bits of STE and our scheme also becomes. We can conclude that our scheme is more efficient than STE in terms of marking bits length.

B. The Number of Packets for Traceback

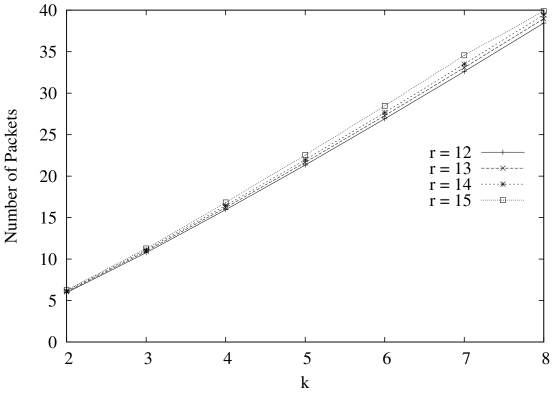

Fig. 5. The number of packets required for traceback

In section IV-B1, in order to evaluate the expected number of packets required for attack path recovery, we have developed an analysis model common to both STE and our scheme. The expected number of packets can be obtained by calculating Eq. (9).

As before, let r be the number of routers on an attack path and k be the number of labels required to represent an interface number. Then we calculate Eq. (9) with r from 12 to 15 and k from 2 to 8. The result is depicted in Fig. 5.

It can be seen from Fig. 5 that in STE and our scheme, the number of needed packets for traceback linearly increases according to k. Furthermore, we see that the number of packets does not greatly depend on r over the range given above. This is partly because the number of packets is computed as the maximum number of labels that some routers put on packets traversing on the attack path.

Anyway the number of packets given in Fig. 5 is rather small and hence we can expect that STE and our scheme work efficiently in practical situations.

VI. Simulation

As discussed in section IV-B1, our analytic model to evaluate the number of packets for traceback is conservative for our scheme. That is, the estimated value of the number of packets just indicates an upper bound of that of our scheme and hence we expect that our scheme should exhibit better performance than STE in terms of the number of required packets. In this section we run simulation experiments to show that our expectation holds.

Let r, p, and k denote the number of routers on an attack path, an initial marking probability, and the maximum number of labels, respectively, as defined in section III. We conduct simulation experiments of our scheme in the following settings: (i) $r = 12, ..., 15$, (ii) $p = 0.4$ and 0.6, and (iii) $k = 2, ..., 8$.

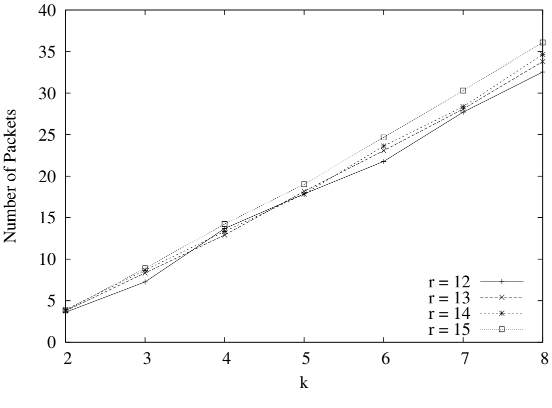

Fig. 6. The number of packets for traceback ($p = 0.4$)

We give the results of the simulations with $p = 0.4$ and 0.6 in Figs. 6 and 7, respectively. From these figures, we readily see that the number of packets required for attack path recovery in each simulation is less than the corresponding result in Fig. 5 in every setting given above. We thus can be convinced that the model given in section IV-B1 is too modest for our scheme and merely gives an upper bound of the number of needed packets for traceback. That is, our scheme usually requires less number of packets to recover attack paths than STE.

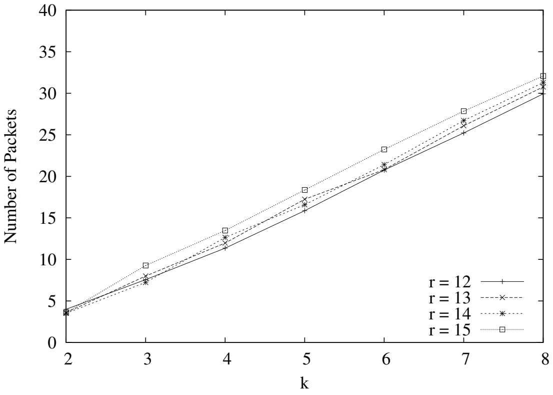

Fig. 7. The number of packets for traceback ($p = 0.6$)

VII. Conclusion

In this paper, we proposed a new, efficient, and adaptive IP traceback scheme. In addition, in this paper we conducted theoretical analysis of the scheme in detail and in particular, we showed that our scheme is more efficient than STE in terms of marking bit length. Finally, we performed simulation experiments of our scheme and showed that in our scheme the number of packets required for attack path recovery is less than STE.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 24500092.

References

[1] M. Muthuprasanna and G. Manimaran, "Space-time encoding scheme for DDoS attack traceback," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Nov. 2005.

[2] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.

[3] D. Dean, M. Franklin, and A. Stubblefield, "An algebraic approach to IP traceback," *ACM Transactions on Information and System Security*, vol. 5, no. 2, pp. 119–137, May 2002.

[4] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP traceback," *ACM/IEEE Transactions on Networking*, vol. 9, no. 3, pp. 226–237, 2001.

[5] S. Yu, W. Zhou, R. Doss, and W. Jia, "Traceback of DDoS attacks using entropy variations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 412–425, Mar. 2011.

[6] M.-H. Yang and M.-C. Yang, "RIHT: A novel hybrid IP traceback scheme," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 789–797, Apr. 2012.

[7] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge Univ Press, 1995.