情報処理学会論文誌 Vol. 51 No. 3 673–681 (Mar. 2010)

Hash ベース IP トレースバックシステムの改良方式の提案と評価

甲斐俊文[†1] 橋口輝[†1] 中谷浩茂[†1]

インターネットの普及にともなって，不正アクセスによる被害が増加傾向にある．特に，送信元アドレスを偽装した DoS（Denial of Service）攻撃や DDoS（Distributed DoS）攻撃は，システムを停止に追いやることもあり，社会生活への影響が出始めている．その対策の 1 つに IP トレースバック技術がある．我々は IP トレースバック技術のうち Hash 方式に注目して研究開発を行っており，既存方式の機能を損なわずに誤検知率を低下させる方式を考案した．本稿では考案した提案方式について，誤検知率とメモリ使用量の関係を表す理論式を示すことで，既存方式と比較して誤検知率が低くなることを示す．また実ネットワーク環境を用いて実施した実験の結果も報告する．

A Proposal and Evaluation of an Improvement Method of Hash-based IP Traceback System

Toshifumi Kai,[†1] Akira Hashiguchi[†1] and Hiroshige Nakatani[†1]

The amount of damage caused by illegal access to networks is increasing with the spread of the Internet. In particular, DoS (Denial of Service) and DDoS (Distributed DoS) attacks with falsified sender addresses can cause systems to stop and this may have a serious impact on society. Various types of IP traceback techniques have been developed and proposed as ways for detecting attackers. We have researched and developed effective IP traceback system for practical use building on existing systems. In this paper, we propose an improved type of the hash-based IP traceback method. We clarify the relationship between the amount of memory and false detection rate and show that the false detection rate is decreased in our hash traceback method. We also test it in a real network environment.

1. はじめに

ネットワークが社会的インフラとして定着した今日，これを用いたサービスの提供は当然のものと認識されている．一方で，それを停止させようとする妨害も増加の一途をたどっている．このような「攻撃」と呼ばれる妨害行為の代表的なものに DoS（Denial of Service）攻撃や DDoS（Distributed DoS）攻撃がある．これらの攻撃は一般的に送信元 IP アドレスを偽装したパケットを用いていることが多く，被害者側には真の発信源や攻撃経路が特定できないため対策が困難なものとなっている．また，DoS 攻撃や DDoS 攻撃以外でも，発信源を偽造して実施される攻撃は存在し，同様の問題がある．

このような真の発信源や攻撃の経路を探査する一般的な手法として，IP トレースバック技術がある[1),2)]．我々は IP トレースバックシステムの実用化を目指して研究開発を進めており，代表的な方式とされる ICMP 方式[3)]，マーキング方式[4),5)]，Hash 方式（ダイジェスト方式）のうち，特に Hash 方式に注目している．

既存の Hash 方式として，BBN Technologies 社の SPIE[6)]，SPIE を基にした Lee らによる Flow トレースバック方式[7)] が提案されている．SPIE は 1 パケット単位での探査が可能であるが，十分な大きさのメモリがなければ誤検知率が高くなる．一方，Flow 方式は同じメモリ量でも SPIE に比べて低い誤検知率で探査ができるが，フロー単位での探査を目的としているため 1 パケット単位での探査はできない．そこで我々は，1 パケット単位で探査が可能で，かつ SPIE に比べて少ないメモリで探査が実現できる方式を考案した．

本稿の 2 章では既存の Hash 方式の概要を述べる．3 章では考案した方式について説明し，SPIE と比較するために誤検知率の理論式を示す．また，実ネットワーク環境で実験した結果について 4 章で示し，5 章で考察を述べる．

2. 既存の Hash 方式について

Hash 方式は，通信路上で通過する個々のパケットから算出した Hash 値をダイジェスト（要約）として記録しておくタイプの IP トレースバック方式である．たとえば**図 1** のように，探査対象パケットについてルータにダイジェストの記録の有無を順次問い合わせることで，パケットの通過経路を特定することができる．なお，Hash 値の記録をルータで行うシ

†1 パナソニック電工株式会社
Panasonic Erectric Works Co., Ltd.

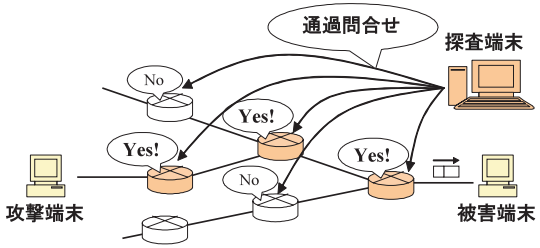

図 1　Hash 方式の経路探査方法
Fig. 1　Hash-based IP traceback system.

ステムのほかにも，外付けのプローブ装置で行うシステムも実現可能である．

パケットをそのまま記録せずに Hash 関数で短くして記録するのは，利用するメモリ量を小さくするためである．しかしそのために異なるパケットの Hash 値が衝突し，通過していない経路で通過ありと判定されてしまう誤検知（フォールスポジティブ）が一定の確率で発生する．

2.1　SPIE 方式

単純に Hash 値をリスト状に記録すると，通過有無を確認する際の検索に時間がかかる．そこで Bloom Filter を用いることにより検索時間を短くしている点が SPIE における Hash 値記録方式の特徴である．通過パケットから算出したビット幅 b [bit] の Hash 値は，ダイジェストテーブルと呼ばれるメモリテーブルに記録される．1 枚のダイジェストテーブルは 2^b [bit] の記録空間を持ち，初期状態ではすべてのビットが 0 になっている．そして**図 2** のようにダイジェストテーブルの先頭から Hash 番目のビットを 0 から 1 にする．探査対象パケットの通過を確認する場合には，確認したいパケットから Hash 値を算出し，ダイジェストテーブルの先頭から Hash 値番目のビットの値を確認する．もしそのビットが 1 であれば通過したと見なし，0 であれば通過していないと見なす．

通過パケットを記録するごとにダイジェストテーブルには 1 が立っているビットが多くなっていく．すると，実際には通過していないパケットの Hash 値が，偶然別のパケットの Hash 値と同値になり，フォールスポジティブが生じやすくなる．そこでダイジェストテーブルを複数枚用意しておき，一定時間ごとにテーブルのローテーションと初期化を行う．n 枚のダイジェストテーブルを t [s] ごとにローテーションさせた場合，$n \cdot t$ [s] の期間のみ通過確認が可能となる．

1 枚のダイジェストテーブルが使うメモリの量 M [bit]，1 枚のダイジェストテーブルに記録されたパケット数 P，そしてフォールスポジティブの発生率 FPR は，次の関係にある．

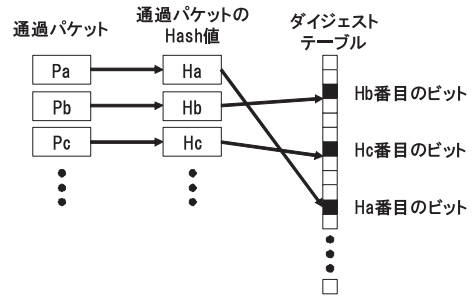

図 2　Bloom Filter による Hash 値の記録方法
Fig. 2　Packet hash recording method of SPIE using Bloom Filter.

- M が同じ値の場合，P が多い方が FPR も大きく，逆に P が少なければ FPR も小さい．
- P が同じ値の場合，M が大きければ FPR は小さく，逆に M が小さければ FPR は大きい．

SPIE では M と P が同じ条件のときでもフォールスポジティブを減少させる工夫として，1 つのパケットから複数個の Hash 値を算出して記録することも提案している．Hash 値を複数算出することでフォールスポジティブが減少するのは，いくつかの Hash 値で Hash 衝突が起きたとしても，複数算出された Hash 値のすべてが Hash 衝突しなければ誤検知とならないためである．ただし，算出された複数の Hash 値は同一のダイジェストテーブルに記録されるため，個々の Hash 値の衝突は生じやすくなる．このため，1 つのパケットから算出する Hash 値の数を大きくしすぎると，個々の Hash 値の衝突率が大きくなり，結果として FPR も大きくなる．

この SPIE 方式のパラメータの関係を数式にすると，以下のようになる．

◎パラメータ
・ダイジェストテーブルのサイズ M [bit]
・記録されたパケット数 P
・フォールスポジティブの発生率 FPR
・1 パケットから算出する Hash 値数 k

① 記録された Hash 値の総数
Pk

② 無関係なパケットから算出した 1 個の Hash 値が，記録された Hash 値うちの 1 個と衝突しない確率

$$1-\frac{1}{M}$$

③ 無関係なパケットから算出した 1 個の Hash 値が，記録された $P\cdot k$ 個の Hash 値のいずれとも衝突しない確率

$$\left(1-\frac{1}{M}\right)^{Pk}$$

④ 無関係なパケットから算出した 1 個の Hash 値が，記録された $P\cdot k$ 個の Hash 値のいずれかと衝突する確率

$$1-\left(1-\frac{1}{M}\right)^{Pk}$$

⑤ 無関係なパケットから算出した k 個の Hash 値がすべて，記録された $P\cdot k$ 個の Hash 値のいずれかと衝突する確率（フォールスポジティブの発生率，FPR）

$$\mathrm{FPR}=\left\{1-\left(1-\frac{1}{M}\right)^{Pk}\right\}^{k} \tag{1}$$

また，SPIE ではネットワーク上の観測点を通過したパケットから Hash 値を算出する際に，**図 3** に示すように，ルータなどで書き換えられる可能性がある部分はゼロで埋めたり，削ったりする．また，IP ヘッダ部分だけでは異なるパケットが同一の値を持つ可能性が高くなることを考慮して，ペイロード部分 64 bit（8 Byte）も含めて Hash 値を算出する．

2.2　Flow トレースバック方式

Lee らが提案している Flow トレースバック方式は SPIE 方式を基にしている．SPIE 方式との違いはパケットを記録する際に同じフローに属しているパケット間で共通の値を持つフィールドだけを対象として Hash 値を求める点である．フローとは一連の通信を意味し，たとえば 1 回の TCP セッションで通信されるパケット群はフローに相当する．IP ヘッダの Source Address や Destination Address，Protocol，TCP や UDP の Source ポート番号や Destination ポート番号などが，フロー内のパケットで共通の値を持つフィールドである．

Flow トレースバック方式では，同じフローに属しているパケットからは同じ Hash 値が算出されることになる．このため，フロー単位での探査は可能であるが，フローに含まれる個々のパケット単位での詳細な探査はできない．その代わり，ダイジェストテーブルに記録

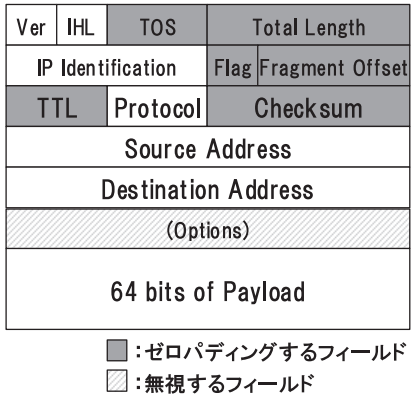

図 3　SPIE 方式で記録する IP パケットフィールド
Fig. 3　The fields of an IP packet using by SPIE.

される Hash 値の数が少なくなり，SPIE よりもフォールスポジティブが発生する可能性が小さくなる．

2.3　既存方式の比較

SPIE 方式と Flow トレースバック方式は Bloom Filter を用いているため，単純にパケットのダイジェストをリストに保存していく方法に比べて短時間で通過確認ができる．

SPIE 方式と Flow トレースバック方式を比べると，機能面では 1 パケット単位での探査ができる点で SPIE 方式が優位である．一方で SPIE 方式は誤検知率を低く抑えるためには大きなメモリ容量が必要とされるが，Flow トレースバック方式はフロー単位での探査に特化することでこの点を改善している．

3. 提 案 方 式

我々は SPIE 方式における複数 Hash 値算出による工夫と，Flow トレースバック方式でダイジェストテーブルに記録される Hash 値を減らす工夫の両方に着目し，1 パケット単位での探査を可能にしたまま SPIE 方式に比べて低い誤検知率で探査できる方式を考案した．

考案した方式では，IP パケットフィールドを 2 種類のグループに分けて，1 つのパケットから複数の Hash 値を算出する際に，それぞれのどちらか一方から算出する．一部の Hash 値は，パケットごとに異なるフィールド（たとえば IP パケットヘッダの ID フィールドや，ペイロード部分など）が含まれているグループから算出する．そして残りの Hash 値は同じ

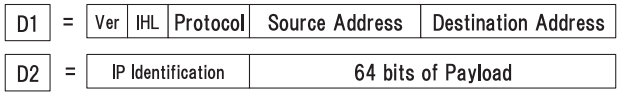

図 4　提案方式で記録する IP パケットフィールド
Fig. 4　The fields of an IP packet using by our method.

フローに属しているパケット間で共通しているフィールドのみで構成されるグループから算出する．こうした組合せにより，1 パケット単位での探査機能を有しつつ，誤検知率を低下させることができると考えた．

以下，3.1 節で提案方式の Hash 値算出方法の詳細を述べ，1 パケット単位で探査が可能であることを示す．3.2 節で誤検知率を数学モデルにより解析し，SPIE と比べて提案方式は誤検知率が低くなることを示す．

3.1　提案方式の Hash 値算出方法

提案方式では，通過パケットから Hash 値を算出して記録する際に，**図 4** に示すように同一フローにおいて共通の値になるフィールドのみを含む部分 D1 と，パケット固有の値を含む部分 D2 とに分けて計算する．D1 と D2 それぞれから最低 1 個 Hash 値を算出していれば，D1 から算出する Hash 値の数と，D2 から算出する Hash 値の数は異なっていてもよい．たとえば合計 5 個の Hash 値を算出する場合に，2 個は D1 から算出し，残りの 3 個は D2 から算出する，ということもできる．

通過の有無を確認する方法は SPIE と同様である．記録時と同じ Hash 関数を用い，指定された探査対象パケットの D1 と D2 から記録時と同数の Hash 値を算出し，そのすべてがダイジェストテーブルに記録されているかどうかにより通過の有無を判定する．

提案方式でも SPIE と同様に 1 パケット単位での探査が可能である．探査時の通過有無の判定は以下のようになされる．

(1)　探査対象パケットがリンクを通過している場合は，記録時にも同じ Hash 値がすべて記録されているので，通過記録ありと判定される．

(2)　探査対象パケットがリンクを通過していない場合は，たとえ同一のフローに属するパケットがそのリンクを通過していても，D2 から算出される Hash 値が異なるため，Hash 衝突がない限り通過なしと判定される．これにより，フロー情報部分を偽造して送信された不審なパケットであっても，ID フィールドやペイロードの値を手がかりにして通過有無を判定できる．

3.2　理論解析による比較

提案方式では SPIE 方式に加えて 2 つのパラメータがフォールスポジティブの発生率に関係する．それは，1 つのパケットから生成される k 個の Hash 値のうち，フロー情報（図 4 の D1）から算出される Hash 値の数 c と，同一テーブル内に同じフローのパケットが記録されている割合 w である．w が大きい環境では，同一のフロー情報を持つパケットの割合が高いため，同じ数のパケットが通過したとしてもダイジェストテーブルに記録される Hash 値の数は少なくなる．このため，w が大きい環境の方がフォールスポジティブの発生率は低くなる．また，c の数が多いほど，w の影響が大きくなる．

提案方式のパラメータの関係式を以下に示す．

◎パラメータ
- ダイジェストテーブルのサイズ M [bit]
- 記録されたパケット数 P
- フォールスポジティブの発生率 FPR
- 1 パケットから算出する Hash 値数 k
- k 個の Hash 値のうち，フロー情報から算出される Hash 値の数 c（$1 \leqq c \leqq k-1$）
- 同一テーブル内に同じフローのパケットが記録されている割合 w

① 記録されたフロー数
$P(1-w)$

② 記録された Hash 値の総数
$P(k-cw)$

③ 無関係なパケットから算出した k 個の Hash 値がすべて，記録された $P \cdot k$ 個の Hash 値のいずれかと衝突する確率（フォールスポジティブの発生率，FPR）

$$\mathrm{FPR} = \left\{1 - \left(1 - \frac{1}{M}\right)^{P(k-cw)}\right\}^{k} \tag{2}$$

SPIE と提案方式の FPR の比較を，**図 5** のグラフで示す．このグラフの例は，1 枚 64 MByte のダイジェストテーブル（$M = 67{,}108{,}864$）で，10 Gbps のリンクを 1 秒間隔で監視した場合の最悪 FPR を示している．なお，10 Gbps の Ethernet が 1 秒間に転送できる最大パケット数はおよそ 15,000,000 パケット（$P = 15{,}000{,}000$）として計算した．

同じフローのパケットが記録されている割合である w の値はネットワークや観測タイミ

ングにより異なるが，TCP のようなセッション型の通信が多く使われていれば，その分だけ w の値は大きくなる．SPIE 方式では w の値は FPR に影響しないが，提案方式は w の値が大きくなると FPR は低下する．このため，SPIE 方式の場合，k の値を最適値（この例では $k = 3$）にしても FPR は 10%以上になるが，提案方式では w が 0.2 程度の環境であっても，k と c の値を最適にすれば FPR は 10%を下回る．また，w が 0.4 の環境なら FPR を約 5%にすることができ，w が 0.6 の環境なら FPR を約 1%にすることができる．

図 6 に，算出される全 Hash 値の数である k を 4 として，c の値を変化させた場合の FPR の変化を示す．D1 から算出される Hash 値は c 個であり，D2 から算出される Hash 値は $k - c$ 個である．w の値にかかわらず，c の値が大きい方が FPR は小さくなる．

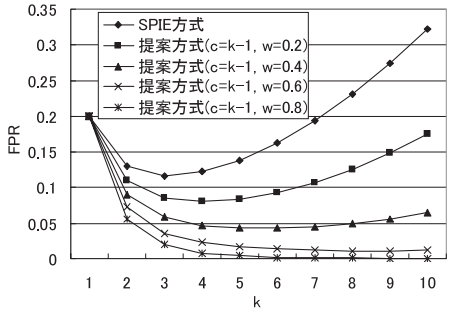

図 5　SPIE 方式と提案方式の FPR
Fig. 5　False positive rates (FPR) of SPIE and our method.

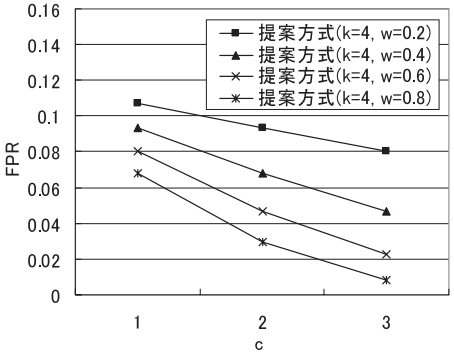

図 6　パラメータ c と FPR の関係
Fig. 6　Relation of parameter c and FPR.

4. 実ネットワークでの実験

提案方式と SPIE の Hash 計算方式を実装したプローブ装置を用いて，実ネットワーク環境でのフォールスポジティブ発生率の測定実験を行った．

4.1 実 験 構 成

実験の構成図を，**図 7** に示す．組織ネットワークのインターネットアクセス回線をプローブ装置で監視することで実験を行った．実際に業務で使用しているネットワークであるためトラフィックの解析は行っていないが，実験前に測定したところ，既設 L3S/W からインターネットに接続される 2 本のリンクには，合計でおよそ 30 Mbps〜50 Mbps のトラフィックが観測された．

この環境に，構成図にあるように 2 カ所の TAP 装置を挟み，それぞれを 2 台のプローブ装置によって観測した．このプローブ装置でリンクを通過するパケットから Hash 値を算出し，ダイジェストテーブルに記録する．L3S/W では Web サーバと接続しているポートについてミラーリングを行い，Web サーバの通信を攻撃パケット特定装置兼マネージャ装置で観測できる構成にした．この構成により，Web サーバの通信に含まれる特定のパケットを攻撃パケット特定装置で検知し，そのパケットが外部ネットワークに接続される 2 本のリンクのうちどちらを通過したものかを Hash トレースバックの仕組みで特定することができる．

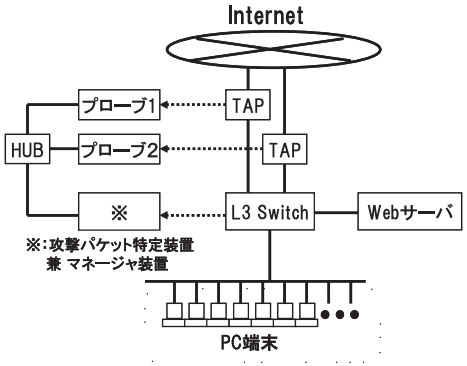

図 7　実験環境の構成
Fig. 7　The system of evaluation test.

なお攻撃パケット特定装置はシグネチャベースの IDS の仕組みで攻撃パケットを検知する．この実験では実際の攻撃を対象とせず，Web サーバに向かって送信される TCP 80 番ポートの syn パケットを攻撃パケットと見なして常時検知するように設定した．トレースバック対象パケットが通過していない方のリンクでフォールスポジティブが発生する割合を測定する実験であるため，攻撃パケット特定装置が検知するパケットのトラフィック量や種類は原理上結果に影響しない．

4.2 実験内容

実験時のシステムの動作は次のとおりである．まず攻撃パケット特定装置は検知したパケットを一定個数集約してマネージャに受け渡す．マネージャは SPIE 方式や提案方式に従ってパケットから Hash 値を算出してプローブ装置に送信し，プローブ装置はダイジェストテーブルをチェックする．1 つのパケットから算出された Hash 値がすべて記録されていた場合にはそのパケットが観測点を通過したものとカウントし，1 つでも記録がなければ通過していないものと判定される．プローブ装置は複数枚のダイジェストテーブルを保持しており，切替え時間ごとにダイジェストテーブルをローテーションさせ，最前面のダイジェストテーブルに Hash 値を記録する．各パケットの通過有無はダイジェストテーブルごとに判定され，通過ありと判定されたパケットの個数がマネージャに報告される．なおパケットごとに通過有無を判定するため，1 回の問合せに同じ Hash 値を持つ 2 つのパケットが集約されている場合でも，通過ありの場合には通過個数 2 としてマネージャに報告される．

本実験ではダイジェストテーブルごとの通過情報をマネージャ装置内のファイルに記録しておき，実験実施中に攻撃パケット特定装置で検知された全パケットのうち，フォールスポジティブやフォールスネガティブ（通過したパケットを通過していないと判定する検知ミス）が発生した割合を調査した．

実験時に使用したパラメータを**表 1** に示す．SPIE 方式と提案方式についてそれぞれダイジェストテーブルの切替え時間を 4 通りに変化させて実験を行った．切替え時間が長いほど，1 枚あたりのテーブルに記録されるパケットの個数が増え，誤検知が増加することになる．各試験はダイジェストテーブルが 20 回切り替わる時間分実施した．

4.3 実験結果

マネージャが記録したダイジェストテーブルごとの通過数報告の例を**表 2** に示す．この例のダイジェストテーブルの切替え時間は 10 s である．また，各プローブ装置はダイジェストテーブルを 10 枚持つため，マネージャからの 1 回の通過確認に対して，それぞれのプローブ装置からの通過報告には 10 テーブル分の通過数情報が含まれる．

表 1　実ネットワーク試験のパラメータ一覧
Table 1　The parameters of evaluation test.

システムパラメータ	試験時の値
ダイジェストテーブルのサイズ(M[bit])	1,048,576 (=128[KByte])
1 パケットから算出する Hash 値数(k)	2
提案方式使用時にフロー情報から算出される Hash 値数(c)	1
ダイジェストテーブルの切替え時間[s]	10, 30, 60, 120
ダイジェストテーブルの枚数	10 枚
攻撃パケット特定装置で集約する対象パケット数	5 packet
Hash 値算出方式	SPIE 方式，提案方式

表 2　ダイジェストテーブルごとの通過数報告例
Table 2　Sample data of attack packet passing number.

通過確認したダイジェストテーブル	プローブ 1 の通過数	プローブ 2 の通過数
90s 前のテーブル	0	0
80s 前のテーブル	0	0
70s 前のテーブル	0	0
60s 前のテーブル	0	0
50s 前のテーブル	0	0
40s 前のテーブル	0	0
30s 前のテーブル	0	0
20s 前のテーブル	0	0
10s 前のテーブル	1	0
最新のテーブル	2	2

フォールスポジティブもフォールスネガティブも発生していない場合，表 2 のように 2 台のプローブ装置の各テーブルで通過が確認されたパケット数の和は，1 回の問合せに含まれる対象パケット数（本実験では 5 packet）と一致する．また，表 2 のように最新のダイジェストテーブルおよび 1 つ過去のダイジェストテーブルでのみ通過が確認される理由は，プローブ装置の観測点を対象パケットが通過してプローブ装置のダイジェストテーブルに記録されてから，攻撃パケット特定装置で検知されてマネージャからプローブ装置へ問合せが行われるまでの時間が，テーブル切替え時間に比べて十分に小さいためである．

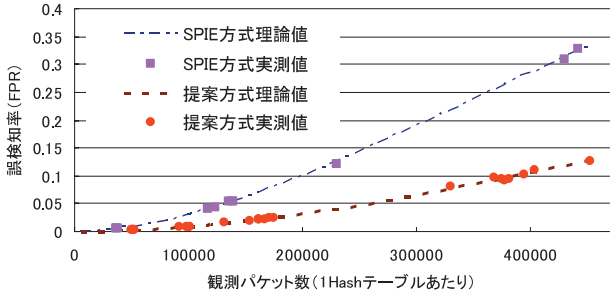

図 8　実験結果と理論計算での FPR の比較
Fig. 8　FPR of evaluation test result and theoretical value.

フォールスネガティブが発生した場合には，2 台のプローブ装置の最新のテーブルと 1 つ過去のテーブルの通過数の和が対象パケット数よりも少なくなる．本実験ではフォールスネガティブは 1 件も発生しなかった．

フォールスポジティブが発生した場合，2 台のプローブ装置の全テーブルの通過数の和が，1 回の問合せに含まれる対象パケット数よりも多くなる．

本実験では次の式 (3) で FPR を算出した．これは 1 つのダイジェストテーブルに対して 1 個のパケットの通過有無を確認した際に，フォールスポジティブが発生した割合を意味する．

$$\mathrm{FPR} = \frac{\text{総フォールスポジティブ発生回数}}{\text{テーブル数} \times \text{集約パケット数} \times \text{問合せ回数}} \tag{3}$$

なお，フォールスポジティブの発生回数を正確に計数するために，最新のテーブルと 1 つ過去のテーブルは分析の対象外とした．したがって分析対象のテーブル数は（プローブ装置の保持テーブル数 − 2）である．総フォールスポジティブ数は，分析対象のテーブルの通過数を，1 回の実験で取得したすべての通過数報告について集計した値である．

この式 (3) により各実験における FPR を算出し，また各実験時間中にそれぞれのプローブ装置が観測した平均 pps（packet per second）とダイジェストテーブルの切替え時間の積により 1 枚のダイジェストテーブルで観測した平均パケット数を算出した．1 枚のダイジェストテーブルあたりの平均観測パケット数を横軸，FPR を縦軸にとり，実験結果と理論式から求めた値を**図 8** に示す．理論式のパラメータ M と k は実験時と同じ値を用い，w は実験時に観測した平均値である 0.93 を用いた．

w の値は環境によって変化するため，この実験結果と理論計算値自体はあくまで一例であるが，SPIE 方式と提案方式ともに，ダイジェストテーブルに記録されたパケット数の大小にかかわらず，理論式と実験結果が一致することを確認できた．

5. 考　　察

5.1　実験結果と数学モデルの差異について

実ネットワーク環境と理論式での誤検知率が乖離する要因として，Hash 計算を行う個所の値がすべて同一なパケットが存在している割合や，算出される Hash 値の偏りについて理論式では考慮していないという点の影響があげられる．ただし，多数の通信が混在する一般的なインターネットトラフィックでは，IP ヘッダとペイロードの先頭 64 bit などがすべて同一のパケットが存在する割合は低く，算出される Hash 値も顕著な偏りなくばらつくと考えられるため，実際の環境で理論式の値のずれは大きくないと予想される．観測を行う環境によって異なるが，少なくとも今回の実験環境ではこうした影響は見られなかった．

5.2　フローに属するパケットの割合について

同じ量のトラフィックであっても，複数パケットからなるフローに属するパケットの割合を意味する w の値が大きいほど，提案方式の誤検知率は低くできる．実際に運用する際には導入前に w の値を測定し，式 (2) を使って最適な k や c の値を設定することが望ましい．なお，本方式を DoS 攻撃や DDoS 攻撃の発信源探査に応用する場合，攻撃パケットが大量に通過するリンクでは，w は平常時と異なる値になる可能性が高い．しかしながら，誤検知率が問題になるのは探査対象パケットが通過していないリンクである．したがって，平常時の w の値を基準にしてパラメータを設定することは妥当な方法と考えられる．

今回の実験環境では w の値は平均 0.93 であったが，現在インターネット通信の多くが TCP で占められていることから，同一セッション内パケットの割合を意味する w の値は，別の環境であっても今回の実験環境と同様に高い値であると考えられる．文献 7) によると，あるリンクの通信を 1 秒間ごとに区切って観測した場合の平均フロー長が 30 packet/flow であったというデータがあり，w に換算すると 0.967 に相当する．図 5 で示したとおり w の値が高い場合，同じメモリ量であっても提案方式は SPIE 方式に比べて顕著に FPR が低くなる．

5.3　パラメータの最適化について

式 (2) から，提案方式では他のパラメータの値にかかわらず c の値が大きいほど FPR が小さくなる．したがって c の最適値は提案方式の原理上の最大値である $k - 1$ である．一方，k の最適値は P と w が明らかであれば，式 (2) を使って導出することが可能である．

P と w は時間や観測するネットワークにより異なるが，プローブで計測することが可能である．したがって k の値を自動的に調整する仕組みも実現可能である．

その場合，ネットワークや時間ごとに k の値が異なるため，通過問合せと通過有無の判定の際に工夫が必要である．たとえば次のやり方で実現できる．マネージャ装置では 1 つのパケットから k のとりうる最大値分の Hash 値を算出して各プローブに問い合わせる．プローブはダイジェストテーブルごとにその時点での k の値を記録しておき，問合せに含まれる Hash 値のうち k 個だけを使用して通過有無の判定を行う．

なお，式 (2) で算出する FPR は，1 パケットを 1 つのダイジェストテーブルに対して確認した結果フォールスポジティブが発生する確率である．実際の運用を考えた場合には，ダイジェストテーブル単位ではなく，プローブ単位で発生するフォールスポジティブが問題となる．対象パケットが通過していない 1 台のプローブへ 1 つの対象パケットを問い合わせた際に，誤って通過ありと判定されるのは，プローブが持つダイジェストテーブルのうち少なくとも 1 枚以上でフォールスポジティブが発生した場合である．この確率を FPRp とし，ダイジェストテーブルの枚数を m とすると，式 (4) により算出することができる．

$$\mathrm{FPRp} = 1 - (1 - \mathrm{FPR})^m \tag{4}$$

式 (4) より，FPR を小さくすると FPRp も小さくなる．したがって式 (2) に基づいてパラメータ c と k を最適化すると，FPRp も最小値にすることができる．

6. おわりに

我々は IP トレースバック技術である Hash 方式について，Hash 値算出方法を工夫することで SPIE よりも誤検知率が低い方式を考案した．監視する通信路において，複数パケットからなるフローに属するパケットの割合が大きいほど，提案方式は SPIE に比べて誤検知率が低くなる．これを理論式により示した．また，SPIE と提案方式を実装したシステムを，実運用されている組織ネットワーク環境に接続し，誤検知率測定実験を行った．これにより，少なくとも実験したネットワーク環境においては理論式と実環境での誤検知率が一致することを確認した．

なお，Hash 方式の IP トレースバックシステムは誤検知率と必要なメモリ量がトレードオフの関係にある．したがって，システムに要求される誤検知率が設定された場合，提案方式を採用すると SPIE に比べて小さいメモリ量で実現することができる．

謝辞　本研究は独立行政法人情報処理研究機構の委託研究「インターネットにおけるトレースバック技術に関する研究」（H17〜H21 年度）による．謹んで感謝の意を表する．

参考文献

1) 門林雄基，大江将史：IP トレースバック技術，情報処理，Vol.12, No.42, pp.1175–1180 (2001).
2) 甲斐俊文，中谷浩茂，清水　弘，鈴木彩子：DDoS 攻撃に対する高性能発信源探査方式の提案，情報処理学会論文誌，Vol.47, No.4, pp.1266–1275 (2006).
3) Bellovin, S.M.: ICMP Traceback Message, InternetDraft:draft-bellovin-itrace-00.txt, submitted (2000).
4) Savege, S., Wtherall, D., Karlin, A. and Anderson, T.: Practical Network Support for IP Traceback, *Proc. 2000 ACM SIGCOMM Conference*, pp.295–306 (2000).
5) Song, D. and Perrig, A.: Advanced and Authenticated Marking Schemes for IP Traceback, *Proc. IEEE INFOCOM 2001*, pp.878–886 (2001).
6) Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Schwartz, B., Kent, S.T. and Strayer, W.T.: Single-Packet IP Traceback, *IEEE/ACM Trans. Networking* (*ToN*), Vol.10, No.6, pp.721–734 (2002).
7) Lee, T.-H., Wu, W.-K. and Huang, T.-Y.W.: Scalable Packet Digesting Schemes for IP Traceback, *Proc. 2004 IEEE International Conference on Communications*, pp.1008–1013 (2004).

(平成 21 年 4 月 7 日受付)
(平成 21 年 12 月 17 日採録)

甲斐　俊文（正会員）
平成 12 年九州工業大学情報工学部知能情報工学科卒業．平成 14 年九州工業大学大学院情報工学研究科博士前期課程修了．同年松下電工株式会社（現パナソニック電工株式会社）入社．トレースバック技術をはじめとするネットワークセキュリティ技術の研究開発に従事．

橋口　輝
平成 15 年工学院大学工学部情報工学科卒業．平成 17 年工学院大学大学院工学研究科修士課程修了．同年松下電工株式会社（現パナソニック電工株式会社）入社．

中谷　浩茂

平成 2 年筑波大学情報学類卒業．同年松下電工株式会社（現パナソニック電工株式会社）入社．LAN 機器相互接続性技術の開発，暗号 ASIC の開発等に従事．現在，未知ウィルス検知をはじめとするネットワークセキュリティ技術の研究開発に従事．