Vol.2010-CSEC-50 No.37
2010/7/2

排他的論理和を用いた Single-Packet IP Traceback

井上　慎一郎[†1,†2]　　石井　方邦[†1]　　笹瀬　巌[†1]

概要：DoS や DDoS 攻撃に対して，その発信元を特定する IP トレースバックは重要な技術である．これまで，マーキング方式やロギング方式が個別に提案されているが，マーキング方式ではトレースバックのために大量の攻撃パケットの収集が必要であるという欠点があり，またロギング方式ではトレースバックサーバによる問い合わせ回数が増大する問題がある．また，２つの方式の欠点を補うためにマーキングとロギングを交互に行う HIT(Hybrid IP Traceback) 方式が提案されているが，トレースバックサーバへの問い合わせ回数が増加したり，悪意ある情報が記入されたパケットによりトレースバックが失敗したりするといった問題がある．そこで本論文では，問い合わせ回数の低減と悪意ある情報の記入によるトレースバック失敗の課題に対して，排他的論理和を用いたトレースバック方式を提案する．本方式では，排他的論理輪を用いることでロギング回数を抑制しトレースバックサーバへの問い合わせ回数の低減が可能である．また，乱数とハッシュ値を用いることで悪意ある情報の記入に対処し，攻撃ホストに繋がるルータの特定を可能とする．計算機シミュレーションにより問い合わせ回数の評価と，故障ルータ数におけるトレースバック成功率を評価し，本方式の有効性を示す．

Single-Packet IP Traceback using Exclusive-Or

Shinichiro Inoue,[†1,†2] Masakuni Ishii[†1]
and Iwao Sasase [†1]

Tracing IP packets back to their origins is an important scheme to defend against denial-of-service(DoS) and distributed-Dos(DDoS) attacks. Mainly, two kinds of IP Traceback schemes have been proposed. One is packet marking which each routers marks their ID into packets. The other is packet logging which each router logs the digest of the forwarded packets. Recently, hybrid IP Traceback scheme, which makes up for shortcoming of above two schemes, has been proposed. Thought his hybrid scheme can reduce overhead on routers rather than conventional schemes, it still take much overhead and moreover the scheme is vulnerable against malicious marking. In this paper, we propose IP Traceback schemes which uses Exclusive-Or. Our scheme can reduce overhead on routers up to two-third. Furthermore, our schemes offer an mechanism to detect malicious marking, and then can decide a exact router which connects to an attacker. By computer simulation, we evaluate an overhead which takes in Traceback process, and Traceback success rate in the case broken routers exist in Internet.

1. はじめに

インターネットは重要な社会インフラの１つとして定着しており，特に企業ではホームページ等が重要な情報発信手段として用いられている．一方，インターネットにおける様々な脅威も存在している．その脅威の１つとして DoS(Denial of Service:サービス拒否) 攻撃，DDoS(Distributed DoS: 分散型サービス拒否) 攻撃が挙げられる．これらは，大量の攻撃パケットを標的サーバに送信することで通信回線やサーバリソースを浪費させ，サービス提供を妨害する攻撃である．しかしながら，攻撃パケットの送信元 IP アドレスは偽装されている場合が多いため，発信元である攻撃ホストを特定することは困難となっている．

そこで，発信元 IP アドレスが偽装されている場合においても攻撃パケットの発信元を特定可能にする技術として IP トレースバックが研究されている．代表的な IP トレースバックとして，中継時にルータが確率的にパケット内にマークを行うマーキング方式[1)2)] や各ルータが中継したパケットのハッシュ値を保持するロギング方式[3)] が提案されている．しかしながら，マーキング方式はトレースバックを行うために大量の攻撃パケットを収集する必要があるという問題がある．また，ロギング方式は，トレースバックを行うために攻撃パケットは１つで十分であるが，ハッシュ値を保持するルータへの問い合わせ回数が増大するという問題がある．

近年，それぞれの問題を補うようにマーキング方式とロギング方式を組み合わせた HIT(Hybrid IP Traceback) 方式[4)] が提案された．HIT 方式では，マーキングとロギングを交互に行うことで，１つの攻撃パケットでトレースバックを行うことが可能となり，更にルータにおいて毎回ロギングを行わず２回に１度ロギングを行うために，ハッシュ値を保

†1 慶應義塾大学理工学部情報工学科
†2 inoue@sasase.ics.keio.ac.jp

Vol.2010-CSEC-50 No.37
2010/7/2

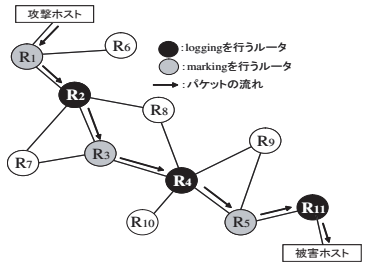

図 1 パケットが被害ホストに至るまでの処理

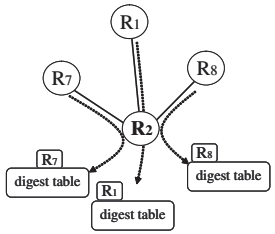

図 2 ルータ R_2 におけるダイジェストテーブル

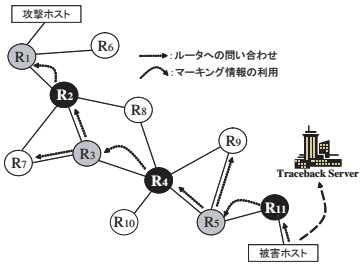

図 3 トレースバック処理

持するルータへの問い合わせ回数を方式 3) と比較し約半分に低減することが可能となっている．しかしながら，２つの課題が存在する．１つ目は，HIT 方式により問い合わせ回数は半減されたものの以前ルータへの問い合わせ回数が多いことである．２つ目は，マーキング方式を用いるため攻撃者に悪意ある情報をパケット内に記入され，トレースバックの失敗を引き起こすということである．そこで本論文では，排他的論理和を用いた Single-Packet IP トレースバック方式を提案する．本方式では，排他的論理輪を用いることでサーバへの問い合わせ回数を低減し，また乱数とハッシュ値を用いることで悪意ある情報の記入に対処する．計算機シミュレーションにより問い合わせ回数の評価と，故障ルータ数におけるトレースバック成功率を評価し，本方式の有効性を示す．

以降，2 章では従来方式である HIT 方式とその課題について述べ，3 章では提案方式である排他的論理輪を用いた Single-Packet IP トレースバックについて述べる．4 章では提案方式の評価を行い，最後に，5 章で本稿をまとめる．

2. 従 来 方 式

本章では，従来方式である HIT 方式とその課題について述べる．HIT 方式はマーキング方式とロギング方式のハイブリッド方式であり，１つの攻撃パケットでトレースバックを行うことが可能である．また，ハッシュ値を保持するルータへの問い合わせ回数を方式[3] と比較し約半分に低減することが可能である．以下，HIT 方式を用いるネットワークおいて，パケットが攻撃者ホストから被害ホストに至るまでの処理方法を 2.1 で述べ，トレースバック処理方法を 2.2 で述べる．そして，HIT 方式の課題を 2.3 で述べる．

2.1 パケットが攻撃ホストから被害ホストに至るまでの処理

本節では，パケットが攻撃者ホストから被害ホストに至るまでの処理方法について述べる．このステップにおいて，各ルータは主に２つの役割を担う．１つは，マーキングであり，もう１つはロギングである．マーキングとは，各ルータが自身の ID(15bit) をパケット内にある identification フィールド (16bit) 中の 15bit に記入し，更に残りの 1bit にマーキングを行った証拠としてフラグを立てパケットの転送を行う処理である．一方，ロギングとは受信したパケットのハッシュ値を算出しダイジェストテーブルに保持し，更にロギングを行った証拠としてフラグを立てパケットを転送する処理である．図 1 にパケットが攻撃ホストから被害ホストに至るまでの例を示す．

初めに，攻撃ホストからパケットを受信したルータ R_1 は自身の ID を記入するマーキングを行い，パケットをルータ R_2 に転送する．次に，ルータ R_2 はルータ R_1 から受信したパケットのハッシュ値を算出し，それを自身のダイジェストテーブルに格納する．ここで図 2 にルータ R_2 におけるダイジェストテーブルを示す．ルータ R_2 は近隣のルータに対するダイジェストテーブルを各々準備している．例えば，ルータ R_1 から受信したパケットのハッシュ値はルータ R_1 用のダイジェストテーブルに，またルータ R_7 から受信したパケットのハッシュ値はルータ R_7 用のダイジェストテーブルに格納するという具合である．以降も同様に，マーキングとロギングを交互に行い，パケットは被害ホストに届く．

2.2 トレースバック処理

本節では，攻撃パケットの発信元を特定するトレースバック処理について述べる．トレースバックには主に２つのプロセスがある．１つは，マーキング情報により通過ルータを特定するプロセスと，もう１つはロギングにより攻撃パケットのハッシュ値を保持するルータを発見することで通過ルータを見つけるプロセスである．図 3 に HIT 方式におけるトレースバック処理を示す．ここで，図 3 のトレースバックサーバとはネットワークトポロジーを把握する第３者機関であり，被害ホストから攻撃パケットを受け取り，実際にトレースバッ

クを行う機関である．図 3 を用いてトレースバック処理を述べる．

初めに，受信したパケットが攻撃パケットであると判断した被害ホストは，そのパケットをトレースバックサーバに送信し，トレースバック処理のリクエストを行う．

次に，トレースバックサーバは受信した攻撃パケット内のフラグより，被害ホストの直前のルータがマーキングを行ったか，またはロギングを行ったかを知ることが可能である．図 3 においては，直前のルータがロギングを行ったことが分かる．そこで，トレースバックサーバは被害ホストの直前ルータであるルータ R_{11} に対して攻撃パケットのハッシュ値を保持しているか，更にその攻撃パケットはどのルータから受信したものであるかの問い合わせを行う．問い合わせを受けたルータ R_{11} は図 2 で示したルータ R_4 のダイジェストテーブルと同様の隣接ルータ用のダイジェストテーブルを保持しており，攻撃パケットのハッシュ値がどのルータ用のテーブルに格納されているかを探査する．ここでは，ルータ R_5 用のテーブルに入っていることから，攻撃パケットはルータ R_5 から受信したものであることが分かり，トレースバックサーバに対してルータ R_5 である情報を返信する．

ルータ R_5 が攻撃パケットの通過ルータであると分かったトレースバックサーバは，ルータ R_5 の全隣接ルータに対して攻撃パケットのハッシュ値を保持していないか問い合わせを行う．以降は，先に述べたプロセスと同様に，図 3 においては，ルータ R_4 が攻撃パケットのハッシュ値を保持していることと，更にそのハッシュ値がルータ R_3 用のダイジェストテーブルに入っているという情報をトレースバックサーバに対して返信する．以上のプロセスを繰り返し，トレースバックサーバは最終的にルータ R_1 にまで辿り着き，トレースバック処理を終了する．

マーキング方式とロギング方式を組み合わせた HIT 方式は，トレースバックを行うためには攻撃パケットが１つで十分である方式である．更に，２つに１つのルータがロギングを行うために，全ルータがロギングを行う方式[3] と比較して問い合わせ回数を約半分に低減することが可能である．

2.3 HIT 方式における課題

本節では，HIT 方式における２つの課題について述べる．

- **ロギングによる問い合わせ回数の増加**
  １つ目の課題は，ロギングを行うために発生する問い合わせ回数の増加である．また，ロギングはハッシュ値を保持する必要があるためルータへの負荷も大きくなる．HIT 方式は方式[3] と比較して問い合わせ回数を半減しているが，トレースパスのホップ数が大きくなると，ロギングを行うルータの数も増加し，結果として問い合わせ回数も増加するという課題がある．

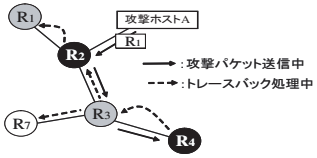

図 4 トレースバックの失敗

- **悪意ある情報の記入によるトレースバックの失敗**
  HIT 方式において，各ルータはパケット内に記入されているフラグにより，マーキングを行うか，またはロギングを行うかを判断している．したがって，攻撃ホストにより悪意ある情報を記入されるとトレースバックの失敗を招くという課題がある．図 4 にトレースバックの失敗例を示す．図 4 において真の攻撃ホストは攻撃ホスト A とする．トレースバックの失敗を招く攻撃例として，初めに攻撃ホストはパケット内のマーキングフラグを立て，更にルータ R_1 の ID を identification 内に記入する．すると，ルータ R_2 はこのパケットがあたかもルータ R_1 から来たものであると判断し，ルータ R_2 はロギングを行う．そして，実際にトレースバック処理の際には，ロギングを行ったルータ R_2 は攻撃パケットがルータ R_1 から来たパケットであるとトレースバックサーバに返信し，R_1 が攻撃ホストに繋がる出発点ルータであると誤った判断がなされ，トレースバックの失敗が発生する．この課題に対処するためには，通常のトレースバック処理により決定された出発点ルータが真の出発点ルータ (攻撃ホストに隣接するルータ) であるのかを検証する仕組みが必要となる．

表 1 従来方式と提案方式の違い

従来方式	[マーキング+ロギング]
提案方式	[マーキング+排他的論理和+ロギング] [出発点ルータ検証機能]

3. 提 案 方 式

本章では，トレースバックサーバによる問い合わせ回数を低減し，またトレースバック処理により特定されたルータが真の出発点ルータであるのかを検証するために，排他的論理和を用いた Single-Packet IP トレースバック方式を提案する．排他的論理輪を用いることで２つのルータの ID を１つにまとめることが可能となり，ロギング回数を抑制しトレースバックサーバへの問い合わせ回数を低減することが可能となる．更に，乱数とハッシュ値を用いることで出発点ルータの検証を可能とし悪意ある情報の記入に対処することが可能となる．表１に従来方式と提案方式の違いを示す．提案方式では，従来方式に排他的論理和を付加した機能と，更に出発点ルータを検証する機能がある．以下では，それら２つ機能を１

Vol.2010-CSEC-50 No.37
2010/7/2

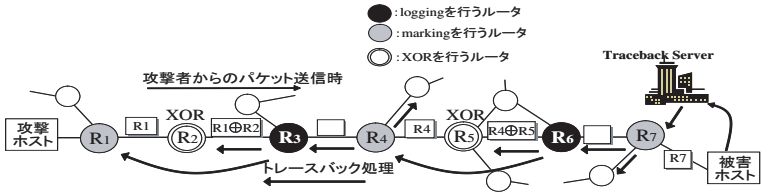

図 5 マーキング+排他的論理和 (XOR)+ロギング

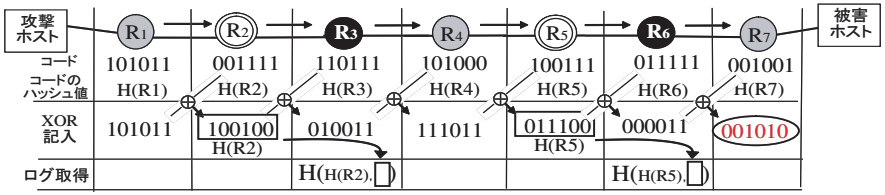

図 6 出発点ルータの検証を行うためのルータ処理

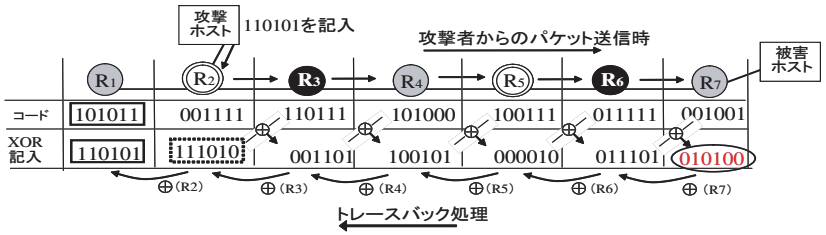

図 7 R_1 が出発点ルータであると判断された場合の検証例

つずつ述べる．

3.1 マーキング+排他的論理和 (XOR)+ロギング

本節では，従来と比較し更に問い合わせ回数を低減するために排他的論理和を用いた方式について述べる．

初めに，パケットが攻撃ホストから被害ホストに至るまでのルータの処理を述べる．各ルータは従来方式と同様にマーキングとロギングを行う上に，更に提案方式では排他的論理を行う．図 5 に排他的論理和を用いたトレースバック方式を示す．図 5 において，初めにルータ R_1 は自身の ID を identification 内に記入するマーキングを行い転送する．そして，ルータ R_2 は自身の ID とルータ R_1 が既に記入している ID との排他的論理輪 ($R_1 \oplus R_2$) を identification 内に上書きし，転送する．次に，パケットを受信したルータ R_3 はパケットのハッシュ値を算出し，自身のダイジェストテーブルに格納する．ここで，提案方式において各ルータが保持するダイジェストテーブルは図 2 のように隣接するルータごとに準備されているものではなく，排他的論理和の値ごとに準備されている．図 5 におけるルータ R_3 には排他的論理輪 ($R_1 \oplus R_2$) のためのダイジェストテーブルがある．以降，同様にしてパケット処理を行い，パケットは被害ホストに届くこととなる．

次に，実際にトレースバックを行う場合について述べる．初めに，被害ホストはトレースバックのリクエストを行うために，攻撃パケットをトレースバックサーバに送信する．図 5 において，トレースバックサーバはパケットのマーキング情報からルータ R_7 を通過ルータであると特定する．そして，トレースバックサーバはルータ R_7 に隣接するルータに対して攻撃パケットのハッシュ値を保持していないかの問い合わせを行う．図 5 では，ルータ R_6 は保持していることと，更にそのハッシュ値が排他的論理輪 ($R_4 \oplus R_5$) 用のダイジェストテーブルに格納されていることから，攻撃パケットがルータ R_4 とルータ R_5 を通過して来たことが分かり，その情報をトレースバックサーバに返信する．以降，同様の処理を繰り返し，ルータ R_1 にまで辿り着くことが可能である．

排他的論理和を用いることで，２つのルータの ID を１つにまとめることができ，ロギング回数を低減することが可能である．従って，トレースバック処理において，ロギングを行うことで必要となる問い合わせ回数を低減することが可能である．

3.2 出発点ルータの検証

攻撃ホストは管理がなされていないルータの ID を悪用することで 2.3 節で紹介したようなトレースバックの失敗を招く攻撃を行うことが可能である．そこで，本節では通常のトレースバック処理によって特定された出発点ルータが真の出発点ルータであるかを検証するため，乱数 (コード) とハッシュ値を用いた検証方式について述べる．ここで，コードとは第３者機関であるトレースバックサーバが管理を行い，各々のルータが保持する非公開の値である．以下では，初めにコードとハッシュ値を用いた方式を述べ，次に本方式を用いた検証例を述べる．

図 6 に，パケットが被害ホストに至るまでの検証機能におけるルータ処理について示す．各ルータはコードとそのハッシュ値を保持しているものとする．初めに，ルータ R_1 は自身のコードをパケットに記入し転送する．パケットを受信したルータ R_2 は自身のコードと記入されているコードの排他的論理輪を算出し，上書きを行う．パケットの転送時には排他的論理輪の値と共に自身のコードのハッシュ値 $\mathrm{H}(R_2)$ を転送する．次に，ルータ R_3 は受信

Vol.2010-CSEC-50 No.37
2010/7/2

した２つの値である排他的論理輪の値と H(R_2) から新たなハッシュ値 H[H(R_2),100100] を算出し，それを保持する．以降のルータも同様の処理を行い，被害ホストには図 6 の丸で囲んだ値 001010 が届く．

次に，上で述べた方式を用いて通常のトレースバック処理によって特定された出発点ルータが真の出発点ルータであるか，またそうでない場合はどのルータが真の出発点ルータであるかの検証例を述べる．図 7 に，ルータ R_2 に繋がる攻撃ホストが通常のトレースバック処理によりルータ R_1 を出発点ルータとして特定され得る攻撃を行っている例を示す．攻撃ホストはコードを記入するフィールドをある値で埋める必要があるため，図 7 においては値 110101 を記入してパケットを送信するとする．被害ホストは攻撃パケットと共に丸で囲んだ値 010100 を受信する．以下，検証プロセスを述べる．

初めに，被害ホストは攻撃パケットと共に，受信した値 010100 をトレースバックサーバに送信する．そして，既に通常のトレースバックによってトレースパスを取得しているトレースバックサーバは，図 7 に示すように，受信した値 010100 と通過ルータのコードとの排他的論理輪を順に算出していく．最終的には通常のトレースバックによって出発点ルータとして特定されたルータ R_1 まで遡る．ここで，算出した値とルータ R_1 のコードが同じ場合は，ルータ R_1 が出発点ルータであると断定することが出来る．しかしながら，その値が異なる場合は出発点ルータは R_1 ではないことが分かる．これは，コードが各ルータとトレースバックサーバの間でのみ共有されている値であり，もし出発点ルータであればその値をパケット内に記入しているためである．図 7 では実線の四角で囲んだ値同士が異なるため，ルータ R_1 は出発点ルータでないことが分かる．従って，ルータ R_2 か R_3 が出発点ルータである．

次に，トレースバックサーバは出発点ルータを特定するために，ルータ R_3 が保持している図 6 で示したハッシュ値の復元を試みる．このハッシュ値の復元には，ルータ R_2 のコードのハッシュ値 H(R_2) と図 7 内の点線で囲んだ値 111010 を用いる．ここで，作成したハッシュ値とルータ R_3 が保持しているハッシュが同じである場合，出発点ルータは R_2 であると断定できる．これは，ルータ R_3 がルータ R_2 のみ知りうるコードから算出されたハッシュ値を持っていることを示している．つまり，ルータ R_3 は攻撃パケットをルータ R_2 から受信したことが証明される．一方，復元した値とルータ R_3 が保持するハッシュ値が異なる場合は出発点ルータはルータ R_3 であることが分かる．値が異なる理由は，ルータ R_3 が保持するハッシュ値がハッシュ値 H(R_2) を用いて作成されたものでないためあり，よって，攻撃パケットはルータ R_2 を通過していないことが証明される．

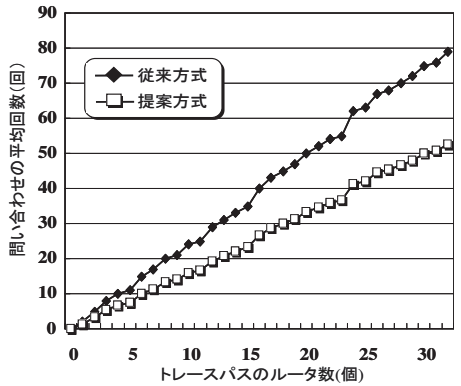

図 8 トレースバックサーバによる問い合わせ回数の評価

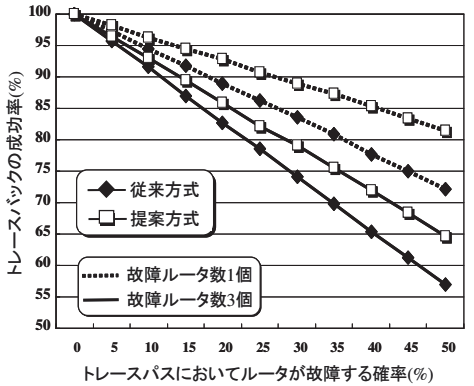

図 9 故障ルータ数におけるトレースバック成功率

本方式を用いることで，従来方式では課題であったトレースバックの失敗を招く攻撃に対処可能であり，真の攻撃ホストに繋がるルータの特定が可能となる．

4. 特性評価

本章では提案方式の有効性を示すために，トレースバックサーバによるルータへの合計問い合わせ回数と，故障ルータがネットワーク内に存在する場合におけるトレースバック成功率の評価を行う．

4.1 トレースバックサーバによる問い合わせ回数の評価

問い合わせ回数が増大すると，トレースバック処理のために生じる通信量や，またトレースバックにかかる時間が増大するため，問い合わせ回数を評価することは有効である．図 8 に，トレースパスの全ルータ数におけるトレースバック処理時に発生した問い合わせ回数を示す．ここで，使用するネットワークトポロジーは ITDK0304[5] である．図 8 に示すように，提案方式は従来方式と比較し，全体として 2/3 低減していることが分かる．これは，排他的論理和を用いて２つのルータの ID を１つにまとめることができ，従ってロギング回数を低減しているためである．従来方式ではルータの２つに１つはロギングを行う方式であるのに対して，提案方式では３つに１つのルータがロギングを行う方式であるためである．問い合わせはパケットのハッシュ値を保持しているルータを探索するために行われるものであり，ロギング回数を低減することで問い合わせ回数の低減を実現している．

4.2 故障ルータの数におけるトレースバック成功率の評価

本節では，トレースバック処理を行う場合において，故障ルータが存在した際のトレースバック成功率を示す．ここで，故障ルータとはトレースバックサーバからの問い合わせに対して応答することが不可能であるルータである．図 9 に，故障ルータ数と故障の発生率におけるトレースバックの成功率を示す．故障ルータ数や故障率がいずれの場合においても提案方式はトレースバック成功率を改善していることが分かる．故障ルータによってトレースバックが失敗する場合は，ロギングを行ったルータが故障している場合である．つまり，トレースバックサーバによる問い合わせに対して，ハッシュ値を保持していることを応答できない場合である．改善された要因としては，上記の評価と同様に，提案方式ではロギング回数を低減しているためであり，故障ルータがロギングを行ったルータである確率が低いためである．

5. 結　　論

本論文では，排他的論理和を用いたトレースバック方式を提案した．本方式では，排他的論理輪を用いて 2 つのルータの ID をまとめることでロギング回数を低減し，結果としてトレースバックサーバからの問い合わせ回数を低減した．また，コードとハッシュ値を用いることで攻撃ホストに繋がる真の出発点ルータを検証するための方式を提案した．計算機シミュレーションにより問い合わせ回数の評価と，故障ルータ数におけるトレースバック成功率を評価し，本方式の有効性を示した．

謝辞

本研究の一部は文部科学省 GlobalCOE プログラム「アクセス空間支援基盤技術の高度国際提携」，および富士通研究所の助成により行われた．関係者各位に深謝する．

参 考 文 献

1) S. Savage, D. Wetherall, A.R Karlin and T. Anderson, “ Practical network support for IP traceback ”, in Proc. ACM SIGCOMM, pp295-306, (2000).
2) 木内忠司, 堀良彰, 櫻井幸一, “ パケット生存時間を用いた確率的パケットマーキングによる IP トレースバック手法の提案 ”, 電子情報通信学会技術研究報告.ISEC, 情報セキュリティ, vol. 108, No.161, pp109-114, July 2008.
3) A .Snoeren, C .Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. kent and W. Strayer, “ Single-Packet IP Traceback ”, IEEE/ACM Trans. Networking, vol. 10, No.6, pp721-734, 2002
4) C .Gong, K .Sarac, “ A More Practical Approach for Single-Packet IP Traceback Using Packet Logging and Marking ”, IEEE Transactions on Parallel and Distributed System, vol. 19, No.10, pp1310-1324, October 2008
5) http://www.caida.org/tools/measurement/skitter/