Vol.2013-CSEC-61 No.31
Vol.2013-IOT-21 No.31
2013/5/10

DDoS攻撃における space-time encoding の効率化

双紙 正和[1]

概要：近年，分散サービス妨害（Distributed Denial of Service, DDoS）攻撃が，インターネットに対する重大な脅威となっている．そのような DDoS 攻撃への有効な対策の一つとして，IP トレースバックがある．ここで，IP トレースバックにおいては，攻撃経路を効率よく符号化することが重要である．そこで本研究では，Muthuprasanna らによって提案された space-time encoding 方式 [7] を元に，より効率のよい方式について検討する．

On Efficient Space-Time Encoding for DDoS Attacks

Masakazu Soshi[1]

Abstract: DDoS (Distributed Denial of Service) attacks have been one of the most serious threats to the Internet security these years. A promising countermeasure against such DDoS attacks is IP traceback schemes. For IP traceback schemes, it is vital to encode attack paths in an efficient manner. In this paper, we propose an efficient space-time encoding by improving the one proposed by Muthuprasanna et al. [7].

1. はじめに

近年，インターネットにおけるセキュリティの脅威が増しているが，その中でも特に深刻な被害をもたらす攻撃の一つとして，サービス妨害攻撃 (Denial of Service Attacks, DoS 攻撃) が挙げられる [6]．これは，攻撃者が，特定のサーバ (攻撃対象ホスト，victim と呼ばれる) に対し大量のパケットを送ることで，そのサーバの機能を停止させる攻撃である．さらに近年では，複数の攻撃者が DoS 攻撃を行う分散サービス妨害攻撃 (Distributed Denial of Service Attack，DDoS 攻撃) が大きな問題となっている [8]．

そこで今まで DDoS 攻撃対策は盛んに研究されてきたが，特に重要な対策法の一つとして，IP トレースバック技術 [1], [4], [5], [9], [10] が研究されている．IP トレースバックでは，攻撃者と victim を結ぶ経路（攻撃経路）上の各ルータが，中継するパケットに関する情報を，そのパケットや自身に記録する．そして，victim とルータがそれらの情報を解析することにより，攻撃経路を復元しようとするものである．

[1] 広島市立大学
Hiroshima City University

このような IP トレースバック方式はこれまで盛んに研究されてきたが，本研究では，特に，ある確率でパケットにマーキングを行い，そのマーキング情報に基づいてトレースバックを行う，確率的パケットマーキング方式を対象にするものとする．

ここで，IP トレースバックを実現する際に重要になってくるのは，攻撃経路の効率よい符号化である．このような符号化の中でも重要なものとして，Muthuprasanna らによって提案された，IP トレースバックのための space-time encoding 方式 [7] があげられる．しかしながら，DDoS 攻撃の際の Muthuprasanna らの space-time encoding は，それほど効率が良いものではない．そこで本研究では，Muthuprasanna らの方式の問題点を指摘し，それよりも効率の良い space-time encoding 方式を提案する．

2. 関連研究

この節では，本研究が対象とする，Muthuprasanna らによる space time encoding [7] について述べる．

2.1 IP トレースバック

space-time encoding は，IP トレースバックを前提とし

た，ネットワークトポロジの符号化である．そこで，ここではまずIP トレースバックについて簡単に述べる．

IP トレースバックは，DDoS 攻撃対策として重要なアプローチである．IP トレースバックでは，攻撃者と victim を結ぶ経路上の各ルータが，中継するパケットに関する情報を，そのパケットや自身に記録する．そして，victim とルータがそれらの情報を解析することにより，攻撃経路を復元することを試みる．

IP トレースバック方式の代表的なものとして，確率的パケットマーキング法が挙げられる．確率的パケットマーキング法（probabilistic packet marking, 以下 PPM と略す）とは，ルータが，受け取ったパケットのヘッダに確率的に攻撃経路の情報を書き込む (マーキングする) 方式である [1], [4], [5], [9].

PPM では，ルータ内部にパケットの情報を保存しておく必要がないため，ルータのストレージ容量に依存しないという利点がある．反面，パケットには断片的な経路の情報しか保存されていないので，攻撃経路を復元するためには，多数のパケットが必要となるという欠点がある．本論文では，IP トレースバックとしては，この確率的パケットマーキング方のみを対象とする．

2.2 Space-Time Encoding Scheme の概要

Muthuprasanna らによる space time encoding [7] は，IP トレースバックの際に，あるルータに接続されている複数のルータの情報を，効率よく符号化するものである．space-time encoding では，ルータとルータを繋げるインターフェースを $0,\ldots,n-1$ と番号付けする[*1]．言い換えれば，各ルータごとに最大 n 個のルータが接続されており (線度が n)，それぞれのルータを，IP アドレスで識別するのではなく，0 から $n-1$ までのインターフェース番号を割り振り，識別するということである．このとき，それぞれのインターフェースは $\lceil \log_2 n \rceil$ ビットで表現できる．

残念ながら，パケットに書き込むことのできる領域は限られており[*2]，すべてのルータのインターフェース情報をパケットにそのまま書きこむことはできないため，$\lceil \log_2 n \rceil$ ビットを m ビットずつの k 個のラベルに分割する．ただし，$m \leq \lceil \log_2 n \rceil$ かつ $k \times m \geq \lceil \log_2 n \rceil$ を満たしていなければならない．

ここで，[7] で考えるパケットマーキングアルゴリズムの概略は以下のとおりである[*3]．なお，以下では，2 進数であることを明記するために，$()_2$ という表記を使うことがある．まず，例として，$n = 16$ とする．すると，$\log_2 n = 4$ であり，ルータに接続するインターフェースすべてを 4 ビットで表現できる．そこで，たとえば，2 ビットのラベル 2 個でそのインターフェースを表現すると仮定する（$m = 2$, $k = 2$）．このとき，たとえば，DoS 攻撃の際に，インターフェース番号が 6（$= (0110)_2$）のルータから，ルータ R に攻撃パケットが送られてくるような状況を考える．すると，$(0110)_2$ というインターフェース番号を，01, 10 という 2 個のラベルに分割して表現することになる．そこで，R を経由して victim に送られる 2 個のパケットに，順に，01, 10 という情報を書き込めば，原理的には victim で，R における 0110 というインターフェース番号を復元できることになる．

2.3 DDoS 攻撃時における Space Time Encoding

残念ながら，DDoS 攻撃においては，2.2 節に述べたような単純な space time encoding は失敗する．このような状況を考えるために，以下の例を考える．まず，2.2 節のように，$n = 16, m = 2, k = 2$ とする．そして，DDoS 攻撃において，インターフェース 6 ($= (0110)_2$)，11 ($= (1011)_2$) から，攻撃パケットがルータ R に送られてくると仮定する．

このとき，R は，インターフェース 6 の 2 個のラベル，01, 10 を 2 個のパケットに書き込んで victim に送り，また，インターフェース 11 の 2 個のラベル，10, 11，別の 2 個のパケットに書き込んで送る．しかしながら，victim は，これら 4 個のラベルを用いて，単純に 2.2 節のように復号することはできない．なぜならば，victim は，たとえば，インターフェース 6 の 1 番目のラベル 01 とインターフェース 11 の 2 番目のラベル 11 を組み合わせて，$(0111)_2 = 7$ というインターフェース番号を復元する可能性があるからである（**図 1** 参照）．

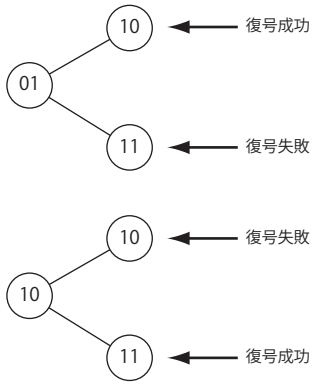

図 1 曖昧な復号
Fig. 1 Ambiguous decoding

以上の議論から分かるように，インターフェース番号を単純に分割するような space time encoding 法は，DDoS 攻撃に対応することができない．

上記のような問題を解決するために，Muthuprasanna ら

[*1] 元の論文 [7] では，インターフェースの番号を $0,\ldots,n$ と仮定しているが，本論文ではより厳密な扱いをするため，このように仮定する．
[*2] [4] によれば，最大 25 ビット程度．
[*3] Muthuprasanna らのパケットマーキングアルゴリズムについては，A.1 節に記した．

は，有限射影平面（より詳しくは，[3], [11] などを参照）の概念を利用した space time encoding を提案した．この基本的なアイディアは，任意の 2 個のラベルが，共通してインターフェース番号のあるビットを含むように符号化するというものである（この結果，すべてのラベルのビット総数は，$\lceil \log_2 n \rceil$ を超える．また，ラベルの大きさは可変になることがある）．このように，ラベル同士が共通するビットを持つことによって，複数のラベルが，同一のインターフェース番号の一部であることをチェックできるようになる．いわば，任意の 2 個のラベルが依存しあっているということもできる．

このような符号化法の例として，$n = 16$ ($\lceil \log_2 n \rceil = 4$) の場合を考えてみる．このとき，4 ビットのインターフェース番号を $b_1b_2b_3b_4$ と書くことにする．この場合の Muthuprasanna らによる space time encoding では，これらの 4 ビットを，b_1b_4, b_2b_4, b_3b_4, $b_1b_2b_3$ という 4 個のラベルに符号化する．こうすると，任意の 2 個のラベルがある 1 ビットを共通に持つようになることに注意せよ．そのような共通するビットでチェックすることによって，victim は一意にインターフェース番号を復号できることになる．

2.4 問題点

2.3 節で述べたような space time encoding 方式の問題点は，符号化の効率が悪いという点である．たとえば，2.3 節で挙げた例では，4 ビットのインターフェース番号 $b_1b_2b_3b_4$ を b_1b_4, b_2b_4, b_3b_4, $b_1b_2b_3$ という 4 個のラベルに符号化するが，このとき，4 ビットを符号化するのに，ラベル全てで $2+2+2+3=9$ ビットが必要になる．

3. 効率の良い Space-Time Encoding の提案

この節では，Muthuprasanna らの方式よりも効率の良い space time encoding 方式を提案する．

3.1 基本的なアイディア

2.4 節で述べたように，Muthuprasanna らによる space time encoding [7] の問題点は，効率が悪い（符号化により多くのビットが必要とされる）という点である．この理由は，有限射影平面を符号化に利用したためである．すなわち，任意の 2 個のラベルが，少なくとも一つのビットを共通して持たないといけないという制約のために，多くのビット長が必要になる．

これを解決する，提案方式の基本的なアイディアは以下のとおりである．そもそも，Muthuprasanna らによる space time encoding では，有限射影平面の制約が強すぎることが問題である．2.3 節で述べたような曖昧な復号ができないよう符号化するためには，単に，インターフェース番号を一意に復号できるような，ラベル相互の依存性があればよい．

そこで，本論文では，インターフェース番号を巡回的に符号化することを提案する．すなわち，たとえば，4 ビットのインターフェース番号 $b_1b_2b_3b_4$ であれば，b_1b_2, b_2b_3, b_3b_4, b_4b_1 の 4 個のラベルとして符号化する．たとえば，インターフェース 6（$=(0110)_2$）は，01, 11, 10, 00 という 4 個のラベルに符号化される．

上記のような提案符号化法においては，Muthuprasanna らによる space time encoding のような，任意の 2 個のラベルにおいて共通するビットがあるという性質はなくなるが，4 個のラベルのすべてのビット割り当てを満たすようなインターフェース番号は，一意に決定できる（**図 2** 参照）．またこの場合，ラベルのビット総数は 8 であり，Muthuprasanna らによる space time encoding より 1 ビット少ない．

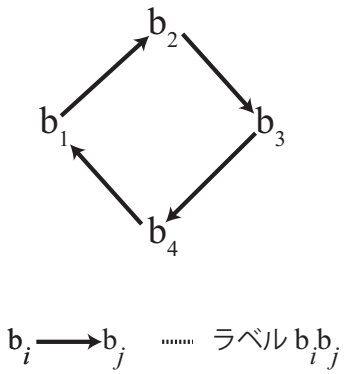

図 2 巡回的な符号化
Fig. 2 Cyclic encoding

表 1 に，2 ビットラベルの場合の，我々が提案する space time encoding 法を示す．表 1 において，列は，インターフェースの数 n の範囲を表し，行は，ラベル番号（送るべきラベルの順番）を表す．また，表記を単純にするため，表 1 の各要素では，ラベル b_ib_j を 単に ij と書いている．そこで，たとえば，表 1 において，「17-32」の列は，n が 17 から 32 の場合（$\lceil \log_2 n \rceil = 5$）を表し，その列の要素に書いてあるような $k=5$ 個のラベルに符号化することを表す．より詳しく言えば，5 ビットのインターフェース番号 $b_1b_2b_3b_4b_5$ を，b_1b_2, b_2b_3, b_3b_4, b_4b_5, b_5b_1 の 5 個のラベルに符号化して，それぞれを一つずつパケットに書き込んで送信することを表している．

なお，[2] によれば，インターネットにおいては，各ルータにおけるインターフェースの総数はほとんどが 128 以下であるため，7 ビットでインターフェース番号を表現することができるが，表 1 では，8 ビット（$n=256$）の場合まで示している．

表 1 提案 space-time encoding（2 ビットラベルの場合）
Table 1 Our space-time encoding with 2 bit labels

	9-16	17-32	33-64	65-128	129-256
1	12	12	12	12	12
2	23	23	23	23	23
3	34	34	34	34	34
4	41	45	45	45	45
5		51	56	56	56
6			61	67	67
7				71	78
8					81

3.2 提案方式

3.1 節で述べた基本的なアイディアをもとに，この節では，一般的な m ビットラベルの space time encoding を提案する．なお，以下では，各ルータのインターフェース数の最大のものを n （インターフェース番号は，0 から $n-1$），また，$L := \lceil \log_2 n \rceil$ とする．さらに，インターフェース番号のビット表現を $b_1 b_2 \cdots b_L$ とおき，$i \le j$ のとき $b[i;\, j] := b_i b_{i+1} \cdots b_j$，それ以外のとき $b[i;\, j] := \epsilon$ （空文字列）とする．また，ビット列 A, B の連接を $A \| B$ と書く．

以降では，提案方式を，n の二つの場合に分けて記述する．

場合 1 ある正の整数 k が存在して，$km-1 < \log_2 n \le km$ のとき

場合 2 場合 1 以外のとき

3.2.1 場合 1 の提案方式

場合 1 では，$\lceil \log_2 \rceil = km$ となるので，

$$k := \frac{\lceil \log_2 n \rceil}{m} = \frac{L}{m} \tag{1}$$

として定義される k は正の整数となる．

このとき，$b_1 b_2 \cdots b_L$ を以下のような，それぞれ m ビットの $2k$ 個のラベルに符号化する．

$$b[im+1;\, (i+1)m] \qquad (0 \le i \le k-1) \tag{2}$$

$$b\left[im + \left\lfloor \frac{m}{2} \right\rfloor + 1;\, (i+1)m + \left\lfloor \frac{m}{2} \right\rfloor\right] \qquad (0 \le i \le k-2) \tag{3}$$

$$b\left[(k-1)m + \left\lfloor \frac{m}{2} \right\rfloor + 1;\, km\right] \| b\left[1;\, \left\lfloor \frac{m}{2} \right\rfloor\right] = b\left[L - \left\lceil \frac{m}{2} \right\rceil + 1;\, L\right] \| b\left[1;\, \left\lfloor \frac{m}{2} \right\rfloor\right] \tag{4}$$

この場合，L（$= \lceil \log_2 n \rceil$）ビットのインターフェース番号が，ラベルすべてで $2L$ ビットに符号化されることになる．

以上で述べた符号化の例を考えるために，たとえば，$n = 250$ とすると，$\lceil \log_2 n \rceil = L = 8$ であり，このとき $m = 4$ とすると，場合 1 に該当する（$k = 2$）．すると，

- 式 (2)　のラベル $(\ell 1), (\ell 2)$

$$(\ell 1)\, b[1;\, 4], (\ell 2)\, b[5;\, 8]$$

- 式 (3)　のラベル $(\ell 3)$

$$(\ell 3)\, b[3;\, 6]$$

- 式 (4)　のラベル $(\ell 4)$

$$(\ell 4)\, b[7;\, 8] \| b[1;\, 2] = b_7 b_8 b_1 b_2$$

という，4 個のラベルに符号化されることが分かる（**図 3** 参照）．

また，$b_1 b_2 b_3 b_4$ における任意の 1 ビットが，2 個のラベルに含まれていることも容易に検証できる．そこで，重複されたビットをチェックすることにより，一意に復号できることが分かる．

3.2.2 場合 2 の提案方式

この場合 2 では，k を以下のように定義する：

$$k := \left\lfloor \frac{\lceil \log_2 n \rceil}{m} \right\rfloor = \left\lfloor \frac{L}{m} \right\rfloor \tag{5}$$

さらに，r を以下のように定義する：

$$r := \lceil \log_2 n \rceil - m \times \left\lfloor \frac{\lceil \log_2 n \rceil}{m} \right\rfloor = L - m \left\lfloor \frac{L}{m} \right\rfloor = L - mk \tag{6}$$

場合 2 においては，$0 < r < m$ となることは明らかである．

このとき，$b_1 b_2 \cdots b_L$ を以下のようなラベルに符号化する．

$$b[im+1;\, (i+1)m] \qquad (0 \le i \le k-1) \tag{7}$$

$$b[im - r + 1;\, (i+1)m - r] \qquad (1 \le i \le k-1) \tag{8}$$

$$b[km - r + 1;\, km + r] = b[L - 2r + 1;\, L] \tag{9}$$

$$b[km+1;\, km+r] \| b[1;\, m-r] = b[L - r + 1;\, L] \| b[1;\, m-r] \tag{10}$$

ここで述べた符号化では，式 (9)　のラベルのみ $2r$ ビットであり，それ以外のラベルは m ビットとなる．したがって，ラベルすべてのビット数は，

$$km + (k-1)m + 2r + m = 2mk + 2r = 2L$$

となり，場合 1 と同様，L ビットのインターフェース番号

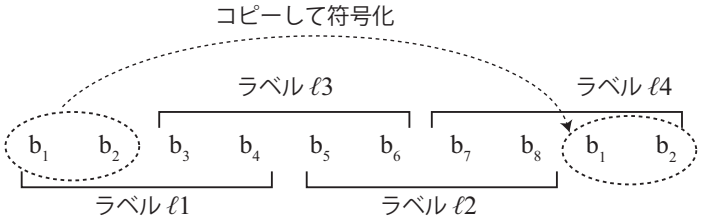

図 3 場合 1 の符号化例
Fig. 3 Encoding example for Case 1

が $2L$ ビットに符号化されることが分かる．

たとえば，$n = 250$ とすると，$\lceil \log_2 n \rceil = L = 8$ であり，このとき $m = 3$ とすると，場合 2 に該当する（$k = 2$, $r = 2$）．すると，

- 式 (7)　のラベル $(\ell 5), (\ell 6)$

$$(\ell 5)\, b\,[1;\, 3]\,, (\ell 6)\, b\,[4;\, 6]$$

- 式 (8)　のラベル $(\ell 7)$

$$(\ell 7)\, b\,[2;\, 4]$$

- 式 (9)　のラベル $(\ell 8)$

$$(\ell 8)\, b\,[5;\, 8]$$

- 式 (10)　のラベル $(\ell 9)$

$$(\ell 9)\, b\,[7;\, 8] \,\|\, b_1 = b_7 b_8 b_1$$

という，5 個のラベルに符号化されることが分かる（**図 4** 参照）．また，任意の 1 ビットが 2 個のラベルに含まれていることも容易に検証できる．

4. 考察

この節では，提案手法について議論する．

Muthuprasanna らによる従来手法 [7] と，特に，2 ビットラベルの場合の提案 space-time encoding 方式について，すべてのラベルのビット総数を，**表 2** に示す．

表 2 従来手法 [7] と提案のビット数の比較
Table 2 Comaprison of the number of total code bits

	9-16	17-32	33-64	65-128	129-256
従来手法 [7]	9	14	18	21	24
提案手法	8	10	12	14	16

表 2 から明らかなように，すべてのラベルのビット総数は，すべての線度の場合について，提案方式の方が小さい．特に，各ルータにおけるインターフェースの総数（線度）が大きくなればなるほど，提案手法の方が効率が良いことが分かる．

また，3.2.1 節および 3.2.2 節で議論したように，場合 1 および場合 2 において，いずれの符号化も，L $(= \lceil \log_2 n \rceil)$ ビットのインターフェース番号が，ラベルすべてで $2L$ ビットに符号化されることが分かる．victim において，攻撃経路の各ルータのインターフェースを一意に復元するためには，インターフェース番号のそれぞれのビットを確認しなければならないことから，L ビットを符号化するためには，$2L$ ビットが少なくとも必要なことが分かる（Muthuprasanna らによる space-time encoding は，それよりも効率が悪い）．したがって，提案方式は，最適な符号化の一つであると考えられる．

また，我々は，2 ビットラベルの場合の，提案手法の予備的なシミュレーション実験も行った．このシミュレーションにおいては，ネットワークトポロジを，victim を根とする 4 分木とした．その木の深さは 2 である（すなわち，ルータの数は 21）．その結果を**表 3** に示す．

表 3 攻撃経路復元のための最小パケット数
Table 3 The smallest number of packets for reconstruction of attack paths

攻撃者数	2	4	8
最小パケット数	7	20	60

また，我々は，その他にも，8, 16 分木の場合についてシミュレーションを行い，経路が効率よく復元できることを確認した．

5. 結論

本論文では，IP トレースバックの際に効率よく攻撃経路を符号化するため，Muthuprasanna らによって提案された space-time encoding [7] を改良した方式を提案した．今後の研究方針としては，インターフェースの総数 n と，パケットに記載できるビット数が与えられた場合，どのような符号化が最も望ましいかについて検討を加える．さらに，インターネットの実際のトポロジに基づいたシミュレーション実験を行い，適切な符号化や性能について考察していくことが考えられる．

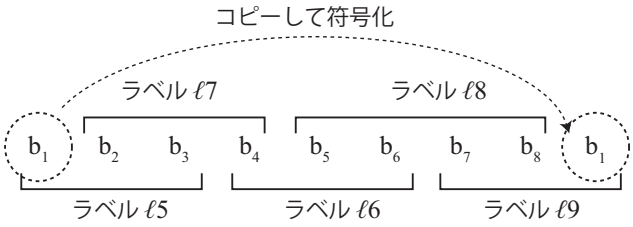

図 4 場合 2 の符号化例
Fig. 4 Encoding example for Case 2

付　　録

A.1 Muthuprasanna らによるパケットマーキングアルゴリズム

この節では，参考のため，Muthuprasanna らによるパケットマーキングアルゴリズム [7] を記す．以下では，各ルータはカウンタ c を持っているものとする．

(1) ルータを起動し，各ルータのラベルカウンタ c を 0 にリセットする．

(2) T を，IP ヘッダー内のマーキング領域ビットとする．

(a) ある確率 p で以下の処理を行う．まず，$c = (c+1) \bmod k$ とする．そして，c 番目のラベルの値を ℓ とし，パケット上に，$1\|c\|\ell$ をマーキングする．ここで，'$\|$' はビット列の連接を表し，1 ビットの値 '1' は，ここでの情報がラベル番号とラベルの値であることを表す．次に，T を右に $\lceil \log_2 k \rceil + m + 1$ ビット，シフトする．

(b) それ以外のとき（確率 $1-p$ で），自らも含め，最も近い上流ルータのラベル番号を c とし（ただし，ラベル番号が書き込まれていない場合は 2a に書かれた処理を行う），c 番目のラベルの値を ℓ とする．$0\|\ell$ をパケットにマーキングする．ここで，0 は，現在のマーキング情報がラベルの値であるということを意味する．次に，T を右に $m+1$ ビット，シフトする．

攻撃経路復元はほぼ明らかであるので，概要のみ示す．victim は，受け取ったパケットから，攻撃経路上におけるマーキング手順を逆順に行うことで，各ルータのラベル情報（インターフェースの一部）を得ることができる．そして，全てのルータの全てのラベル番号を受け取ったとき，経路を復元することが可能となる．詳細については [7] を参照されたい．

参考文献

[1] Adler, M.: Tradeoffs in Probabilistic Packet Marking for IP Traceback, *Proceedings of 34th ACM Symposium on Theory of Computing (STOC)*, Montreal, Quebec, Canada, ACM Press, pp. 407–418 (2002).

[2] Al-Duwairi, B. and Daniels, T. E.: Topology based Packet Marking, *Proceedings of 13th International Conference on Computer Communications and Networks (ICCCN) 2004*, pp. 146–151 (2004).

[3] Colbourn, C. J. and Dinitz, J. H.(eds.): *Handbook of Combinatorial Designs*, Chapman and Hall/CRC, 2nd edition (2006).

[4] Dean, D., Franklin, M. and Stubblefield, A.: An algebraic approach to IP traceback, *ACM Transactions on Information and System Security*, Vol. 5, No. 2, pp. 119–137 (2002).

[5] Karasawa, T., Soshi, M. and Miyaji, A.: A Novel Hybrid IP Traceback Scheme with Packet Counters, *The 5th International Conference on Internet and Dsitributed Computing Systems (IDCS 2012)*, Lecture Notes in Computer Science, Vol. 7646, Springer-Verlag, pp. 71–84 (2012).

[6] Mirkovic, J., Dietrich, S., Dittrich, D. and Reiher, P.: *Internet Denial of Service: Attack and Defense Mechanisms*, Prentice Hall (2005).

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

[8] Peng, T., Leckie, C. and Ramamohanarao, K.: Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS problems, *ACM Computing Surveys*, Vol. 39, No. 1 (2007).

[9] Savage, S., Wetherall, D., Karlin, A. R. and Anderson, T.: Practical network support for IP traceback, *Proceedings of the ACM SIGCOMM*, pp. 295–306 (2000).

[10] Snoeren, A. C., Partridge, C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Kent, S. T. and Strayer, W. T.: Hash-Based IP Traceback, *Proceedings of the ACM SIGCOMM*, pp. 3–14 (2001).

[11] Stinson, D. R.: *Combinatorial Designs: Construction and Analysis*, Springer-Verlag (2004).