社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－DPS－117 (23)
2004－CSEC－ 24 (23)
2004／3／5

コルモグロフ・コンプレキシティの複数の近似算出法とサービス妨害検知パフォーマンス

古谷 隆行　松浦 幹太　アンダーソン・ナシメント　今井 秀樹

東京大学生産技術研究所

和文抄録 昨今、ネットワークセキュリティにおいて、可用性の面から、サービス妨害攻撃 (DoS) が問題になっている。DoS 攻撃は、リモートホストやネットワークのリソースを消費させ、悪意のないユーザへのサービスの質を低下させる攻撃である。本研究では、情報理論的な Kolmogorov Complexity という概念により DoS 攻撃の検知を試みた研究を振り返り、より効率的で精密度の高い DoS 攻撃の検知を行うため、バイナリ文字列間の相関を異なる手法で推定している。シミュレーションを行うことで、Kolmogorov Complexity の DoS 攻撃検知への有効性を提示する。

Estimations of Kolmogorov Complexity and Detection Performance of Denial-of-Service Attacks

Takayuki Furuya　Kanta Matsuura　Anderson Nascimento　Hideki Imai

Institute of Industrial Science, the University of Tokyo

Abstract These days, the Internet accessibility makes it vulnerable in terms of security. Above all, Denial-of-Service (DoS) attacks have become one of the most serious threats. DoS attacks consume a network or remote host's resources and degrade service to legitimate users. Recently, it has been tried to introduce 'Kolmogorov Complexity' into a method to detect DoS attacks.Kolmogorov Complexity is a concept to measure the size of the smallest program capable of representing the given piece of data. In our study,we present availability of applying Kolmogorov Complexity to detect DoS attacks by introducing some estimations of Kolmogorov Complexity.

1 はじめに

近年、ネットワークセキュリティにおいて、Denial-of-Service(DoS) 攻撃、更には複数の攻撃元から DoS 攻撃を仕掛ける分散型 DoS(Distributed Denial-of-Service: DDoS) 攻撃が問題視されている [1], [2]。DoS とは、サーバアプリケーションやプロトコルの特性の脆弱性につけ込むことなどにより、攻撃の対象とするサーバを、極端な過負荷状態や異常状態に陥れることで、サーバが提供するサービスを妨害する攻撃と理解することができる [3], [4]。この DoS 攻撃に対し、経験則に基づいた観点から、これまで様々な対策が提案されている。

経験則に基づく DoS 対策の方針は、IP traceback[5] の様に DoS 攻撃を受けた後でその攻撃者を特定する方針と、DoS 攻撃を検知してフィルタリングなどの対処を施す方針の、二種類に大別することができる。攻撃を検知する従来の対策法方針としては、ingress filtering[6] のようにソース IP アドレスで判断する手法、シグナチャを用いる手法などがこれまでに提案されている。これらの検知手法には、deployment problem や false negative が発生するといった問題点がある。これらの問題は侵入検知技術一般にみられることであって [7]–[9]、(D)DoS 検知ゆえに顕在化しているとは限らないが、(D)DoS 検知を意識して最適化する研究が十分なされているとも言い切れない。(D)DoS 攻撃の場合、その広範さ [10] ゆえ、検知側も複数手法を協調動作させることも選択肢に入れるのであれば [11], [12]、(D)DoS 検知により適した手法を検討し、そのチューニングを行っていくことも重要な研究テーマとなる。

ネットワーク技術的な経験則が目立つ中、(D)DoS 攻撃の広範さに対してメトリックの汎用性で対抗する情報理論的アプローチを採っているという意味で注目に値する手法として、Kolmogorov Complexity を用いた検知手法 [13] が提案されている。Kolmogorov Complexity は、与えられたデータを記述するプログラムのうち最小のも

ののサイズとして理解できるが、実際に求めるには何らかの推定手法を考案して計算する必要がある。既存の検知手法 [13] は、「Kolmogorov Complexity により、与えられたデータの恣意性を測定することができるので、行われている通信が正規のものであるのか攻撃であるのかを判断することができるのではないか」という発想に基づいている。しかし、その従来研究では、サンプリング手法や、その手順などの記述が不足しており、DoS 攻撃検知のパフォーマンス向上について考察がなされていない。

本研究では、従来手法 [13] と異なる Kolmogorov Complexity 推定手法を提示し、実機を用いたネットワークによる実験における振る舞いを比較した。本稿では、以下の手順でその実験の報告を行う。2 章で DoS 攻撃と Kolmogorov Complexity を説明し、3 章で [13] における従来手法、Lempel Ziv を用いた手法及びそれまでとは異なる計算法による DoS 攻撃検知法について述べる。4 章で、実験を行った環境を述べ、5 章で実験的比較とその結果についての考察を行う。最後に、6 章で今後の課題点などをまとめる。

2 背景知識

本章では、DoS 攻撃と Kolmogorov Complexity についての説明を行う。

2.1 Denial-of-Service(DoS) 攻撃

DoS 攻撃は、コンピュータの CPU やメモリリソースに膨大な負荷をかけることにより、悪意のないユーザへのサービスの提供を妨害する。複数の攻撃元から、一斉に DoS 攻撃を行うことでネットワークコンピュータに過剰な負荷をかける DDoS(Distributed Denial-of-Service) 攻撃は、さらに大きな問題になっている。

前章で述べた通り、DoS 攻撃対策としては、攻撃された後、その攻撃を行った者を特定する方法と、攻撃の検知やフィルタリングを行う方法などが研究されている。このうち、本研究は後者に焦点を当てたものであると言える。

本研究は、フィルタリングではなく、攻撃を検知することを最終的な目的としている。被害者に近い場所で、異常な量のパケットが送信される DoS 攻撃を、情報理論学的な Kolmogorov Complexity という概念を応用することによって、検知することを試みている。

2.2 Kolmogorov Complexity

Kolmogorov Complexity とは、前章で述べた通り、データを記述するプログラムのうち最小のもののサイズと理解できる。Kolmogorov Complexity の定義は以下の通りである [14]。

$$K_{\mathcal{U}}(x) = \min_{p:\mathcal{U}(p)=x} l(p)$$

ここで、x は有限なバイナリ文字列、$\mathcal{U}$ は universal computer、$l(x)$ は文字列 x の長さ、$\mathcal{U}(p)$ は、プログラム p で表される際のコンピュータ $\mathcal{U}$ の出力である。

これによって、恣意性を測定することができる理由を、完全にランダムな文字列を例にとって説明する。このようなランダムな文字列を生成するプログラムの長さは、その文字列の長さに完全に依存すると考えられる。

つまり、逆にそれ以外の場合は、文字列を生成するプログラムは、文字列がランダムである場合と比較すると、短くなる筈であると考えることができる。この考え方は、あるバイナリ文字列 X に対して、その Kolmogorov Complexity を $K(X)$ と表すことにすると、次の公式で表すことができる。

$$K(X_1, X_2) \leq K(X_1) + K(X_2) + \log n + c \quad (1)$$

ここで、$K(X_1, X_2)$ は、二つのバイナリ文字列 X_1 と X_2 を連結させた文字列に対する、Kolmogorov Complexity であり、n は文字列 X_1, X_2 の長さである。

(1) 式の証明は [15] を参照してほしいが、成り立つ理由を例を挙げて説明する。二つの文字列 X_1, X_2 が次のような場合を考える。

$$\begin{cases} X_1 = 10010110101110... \\ X_2 = 01101001010001... \end{cases}$$

この場合、文字列 X_1, X_2 はそれぞれランダムであるが、互いに相関があると考えることができるので、(1) 式の右辺は X_1, X_2 の長さに依存するのだが、左辺の X_2 の部分は X_1 との相関により、プログラム内で記述することができるため、右辺よりも Kolmogorov Complexity の値は小さくなるのである。

DoS 攻撃を検知する際は、(1) 式で左辺が非常に小さくなる際に、DoS 攻撃が行われていると判断する。次章で、[13] の研究でどのようにこの Kolmogorov Complexity の概念が DoS 攻撃の検知に応用されたかを説明することにする。

3 Kolmogorov Complexity の推定による DoS 攻撃検知

3.1 確率論を用いた Kolmogorov Complexity の推定法

[13] の研究においては、(1) 式において左辺の値が右辺の値より著しく小さくなった場合に、DoS 攻撃が行われ

ていると判断している。これは、よく似た大量のパケットを、互いに離れた複数の場所から攻撃対象へ一斉に送信した際、トラフィックパターンに大きな類似性がみられると考えられたからである。逆に、正規の通信ではトラフィックパターンは多様性をもち、(1) 式において X_1 と X_2 は相関性を持たないため、左辺と右辺の値が大きく異なることがないという結果が得られる。

このため、パケットを定期的にサンプリングし、プローブで以下の differential を計算する。

$$\{K(X_1)+K(X_2)+...+K(X_m)\}-K(X_1X_2...X_m) \quad (2)$$

これがほぼ 0 であれば正規の通信と考えられるが、パケット間で高い相関関係があると、データ列 $X_1X_2...X_m$ はより小規模なプログラムで表現されるので、(2) 式の値は大きくなる。

Kolmogorov Complexity は計算することができないため、様々な推定法が存在する。[13] の研究では以下のように Kolmogorov Complexity を推定している。

$$\hat{K}(X) \approx l(X)H(\frac{x\#1}{x\#1+x\#0}) + \log_2\{l(X)\} \quad (3)$$

$$H(p) = -p\log_2 p - (1-p)\log_2(1-p) \quad (4)$$

x#1,x#0 はそれぞれ、データ列 X の中の 1,0 のビット数である。

Kolmogorov Complexity を (3) 式のように定義しており、(3) 式で用いるエントロピーを (4) 式のように定義している。

この推定法には問題点がある。Kolmogorov Complexity を推定するのに、確率論を用いているので、例えばデータ列 X が、1010101010101010 である場合と、それと同じ数だけ 1 と 0 がランダムに並んでいる場合で Kolmogorov Complexity の値が全く等しくなってしまう点である。

3.2 Lempel Ziv 78 を用いた推定法

本研究では、Lempel Ziv 78 を Kolmogorov Complexity の推定に用いた研究 [16] を、実際の DoS 攻撃の検知にどの程度有効であったかについても、結果を出している。

Lempel Ziv 78 は直感的には、文字列の中で以前出てきた部分文字列のパターンが出てきたらその部分は圧縮できる、即ち、その部分のプログラム記述は、以前そのパターンが出ていた部分を参照することができるという考えに基づいている。例えば、次の様な文字列があったとする。

10110100100110100100111010010011000 10

この文字列に対して、前から順にそれまでに出てきていない部分文字列毎に区分けすると次のようになる。

1,0,11,01,00,10,011,010,0100,111,01001,001,100,010

部分文字列の数が少なく、文字列として長いものが多くなるほど、以前出てきた部分文字列との類似度が高いと考えられる。このため、その部分文字列を記述する際は、以前出てきた文字列を記述した部分を参照することができる。このため、文字列全体を記述するプログラムのサイズは小さくなり、Kolmogorov Complexity の値は小さくなる。Lempel Ziv は改良された計算法も提案されているので、[16] を参照してほしいが、本研究では (5) 式の様に Kolmogorov Complexity を推定している。l は、部分文字列の数である。

$$\hat{K}(X) \approx l\{2+\log_2(l-2)\} \quad (5)$$

3.3 本研究における DoS 攻撃対策法

トラフィックパターンの相関を調べるためには、同じフロー内で複数個のパケットをサンプリングし、データ列自体の相関関係を調べる必要がある。二つのパケット間での相関関係を調べることを考える。ここで、パケットのデータ長を n とし、パケットを前半と後半に分け、その二つのデータで各ビット毎に EQUIV をとった値を全部足した値を n_{equiv} とする。本研究で採用している、トラフィックパターンの相関を調べる手法は次の通りである。

$$\hat{K}(X) \approx nP(\frac{n_{equiv}}{\frac{n}{2}}) + \log_2 n \quad (6)$$

$$P(y) = \begin{cases} 2y & \text{if } 0 \le y \le \frac{1}{2} \\ 2-2y & \text{if } \frac{1}{2} \le y \le 1 \end{cases}$$

比較する二つのデータを X_1, X_2 とすると、n_{equiv} は X_2 が X_1, もしくは X_1 の各ビットを反転させた $\bar{X_1}$ にどれだけ近いのかを示しているため、(6) 式により複数個のデータの相関関係を求められるのではないかと考えることができる。

4 実験環境

4.1 DoS 検知への Kolmogorov Complexity の応用

Kolmogorov Complexity を用いて DoS 攻撃を検知する流れを説明する。

まず、一つのフロー内でパケットをサンプリングし、各パケット $(X_1, X_2, ...)$ において Kolmogorov Complexity を計算し、その後、複数のパケット列を連結させたバイナリ文字列に対し、Kolmogorov Complexity を計算する。

計算したそれらの値を用いて、(2)式の様に、Kolmogorov Complexity の differential の値を求める。

DoS 攻撃を検知する際は、この differential の値が、実験的に得られた閾値以上であれば、DoS 攻撃と判断する様にする。これは、一つのプログラムから各パケット $X_1, X_2, ..., X_m$ が出されたのであれば、(2) 式で求めた differential の値が大きくなると考えられるからである。逆に、(2) 式で求めた値が、実験的に得られた閾値より小さければ、正規のトラフィックと判断する。

[13] には、追実験できるような情報が十分に書かれておらず、検討すべき具体的手順の記述が不十分であるために、そこで使われているネットワーク及び戦略が十分 DoS 攻撃の検知にチューニングされているのかどうか疑わしい。このため、本研究では、[13] における Kolmogorov Complexity の推定法により得られる結果と、Lempel Ziv による推定法から得られる結果と、本研究における推定法により得られる結果を比較しながら、どの計算法が DoS 攻撃の検知に応用できるのかについて実験及び考察を行っている。

4.2 実験に用いたネットワーク

トポロジーは図 1 の様に設計してある。

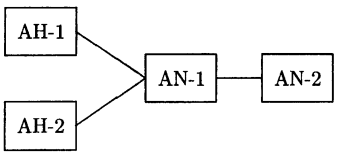

図 1: [13] で利用したトポロジー

[13] では、このトポロジーを用いて、ノード AH-2 からノード AN-2 へ攻撃フローのみを流した場合と、その攻撃フローに加え、ノード AH-1 からノード AN-2 へ音声フロー (正規の通信) を流した場合で、結果を比較している。パケット数を数えることで判断する方法と、Kolmogorov Complexity を推定する方法で、きちんとこの二通りのトラフィックを区別することができるかどうか (正規の通信が攻撃フローと同じノードを通るようでも、両者を区別できるか) について実験が行われている。

DDoS 攻撃を検知するのであれば、ルータに流れ込む、複数のフロー間の相関関係を調べる必要があると考えられる。しかし、本研究では DoS 攻撃を検知できるかに焦点を当てているので、一つのフロー内において、あるパケットが他のパケットとどのような相関関係を持っているのかを調べればよい。

[13] の研究では、正規のフロー内における、パケット間の相関関係が調べられていないので、本稿ではその結果も次章に載せることにする。

実験には、[13] の研究と同様、図 1 のトポロジーを用いている。ノード AH-2 から攻撃フローのみを、AN-1 を通し AN-2 に通す場合と、ノード AH-1 から正規の通信のみを行った場合と、これら二つの通信を同時に行った場合において、Kolmogorov Complexity を用いて DoS 攻撃が検知できているかどうかについて実験を行っている。

ターゲットとなる、AN-2 のマシンを DoS 攻撃から守ることを想定しているので、ノード AN-1 において、DoS 攻撃の検知を行うことになる。そのため、単純化のため、AN-1,AH-1,AH-2 の 3 つのマシンから成るネットワークを構成して実験を行っている。

4.3 パケットのサンプリング

ノード AH-2 から AN-1 に向けて流す攻撃フローとしては、今回 Syn flood を用いている。ツールとしては、Syn flood のシミュレーションに頻繁に用いられる synk4 を用いた。他の攻撃を用いた DoS 攻撃の検知を試みることが今後の課題点に挙げられるが、本研究では、これ以降は Syn flood を検知に用いる DoS 攻撃として話を進めることにする。

また、正規の通信には、AN-1 において apache を用いて web サーバを立ち上げ、AH-1 において、そのサイトを閲覧するという操作を採用している。

データを集めるのに、攻撃フローのみを流す場合においても、正規の通信のみを行う場合においても、両方の通信を行う場合においても、20 秒間パケットのサンプリングを行った。パケットのサンプリングの間隔は、0.1〜1 秒に設定して、サンプリングを行っている。したがって、0.1 秒でパケットをサンプリングした場合は 200 個、1 秒でパケットをサンプリングした場合は 20 個データが得られている。本研究では、サンプリング間隔を変化させつつ、どの計算法で DoS 攻撃を検知する有効性を見出すことができるのかについて実験を行っている。

したがって、本稿では、(2) 式のように、Kolmogorov Complexity の差異を求めるのに、2 個のパケット間でのみ、実験を行っている。最終的な差異の値としては、各々のサンプリング間隔において得られた差異の総平均を採用している。パケットの抽出には、WinDump を用いて行っている。

5 実験結果

実験結果は、以下のものを提示する。

- [13] における Kolmogorov Complexity の推定法を用いた場合
  - DoS 攻撃フロー (Syn flood) のみを流した場合
  - 正規の通信のみを行った場合
  - 両方の通信を行った場合

- Lempel Ziv による推定法を用いた場合
  - DoS 攻撃フロー (Syn flood) のみを流した場合
  - 正規の通信のみを行った場合
  - 両方の通信を行った場合
- 本研究における推定法を用いた場合
  - DoS 攻撃フロー (Syn flood) のみを流した場合
  - 正規の通信のみを行った場合
  - 両方の通信を行った場合

前述の実験環境において、[13] の研究における、Kolmogorov Complexity の推定法に基づき、DoS 攻撃の検知を試みた結果は以下の通りである。

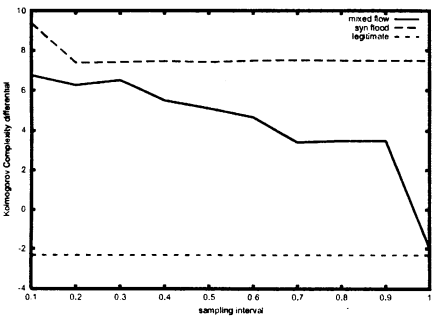

図 2: [13] における推定法による結果

横軸は秒単位でサンプリング間隔、縦軸は (2) 式で求めた Kolmogorov Complexity の差異として、図 2 を表している。3 通りにおいて、Kolmogorov Complexity の差異の値はそれほど変わらない値を示している。正規の通信においては、Syn flood に比べ、パケットのやりとりがそれほど頻繁に行われない。そのため、正規の通信では、図 1 において、ノード AH-2 からノード AN-1 に向けて送信されたパケットについて、サンプリングを行わずに逐一 Kolmogorov Complexity の差異を求め、その総平均の値を、図 2 の中で、点線で示している。

正規の通信と Syn flood が両方ともなされている場合において、差異を求めると、図 2 のように、サンプリング間隔を大きくすると共に正規の通信と見なされ易い傾向になることが分かった。しかし、図 2 程度の差異の差では、実験環境を変えるだけで、DoS 攻撃の検知に必要なサンプリング間隔の値が変化してしまう。

Lempel Ziv による推定法の結果は図 3 の通りである。

図 2 の場合と異なり、サンプリング間隔と推定値の間に相関関係が見られない。正規の通信が行われている背後に DoS 攻撃が加えられたとしても、この推定法では、サンプリング間隔を変えることによって、DoS 攻撃を検知することは不可能である。

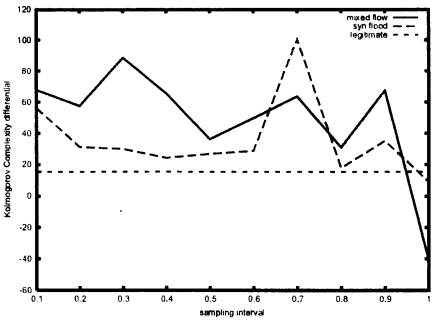

図 3: Lempel Ziv を用いた推定法による結果

次に、本研究における推定法に基づき、DoS 攻撃の検知を試みた結果は図 4 の通りである。

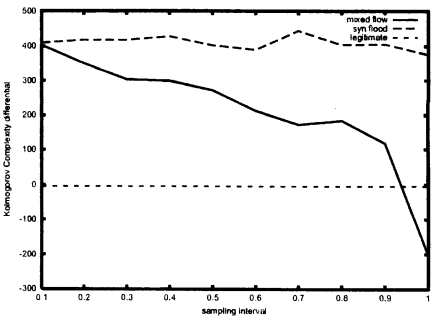

図 4: 本研究における推定法による結果

図 4 においても、図 2 と同様、Syn flood のみでは Kolmogorov Complexity の差異の値は、サンプリング間隔によって大きく異なることはないが、正規の通信と Syn flood が交じり合ったトラフィックが流れる場合は、サンプリング間隔が大きくなるほど、正規の通信と見なされてしまうことが示されている。図 2 と比べると、Kolmogorov Complexity の差異の値が Syn flood のみの場合と、両者交じり合ったトラフィックの場合でかなり大きくなっているのが分かる。これは、用いているデータは全く同一のものであるため、推定法の違いによるものである。図 2 と、図 4 で、定性的には同じであり、DoS 攻撃を検知する性能としては明確な差がないことが分かった。しかし、同じ性能が出るのであれば、定義から理論的に DoS 検知に適していると考えられる本手法の方が良いのではないかと考えられる。

また、本研究において、長期に渡り、Kolmogorov Complexity の差異を測定し続けることを仮想するネットワークにおいては、サンプリング間隔を一定にしておくことで、Kolmogorov Complexity の差異の変化を観

測することにより、DoS攻撃を検知することができる可能性もあるという、新しい知見を見出すこともできた。

6 結論及び今後の課題

本稿では、既存のKolmogorov Complexity推定法に基づくDoS攻撃の検知の結果と、Lempel Zivを用いた場合の結果と、本研究における推定法に基づくDoS攻撃の検知の結果を比較し、どの推定法でDoS攻撃を検知する有効性が示されるのかについて、実験を行った。

前述の通り、本研究では(2)式のように、Kolmogorov Complexityの差異を求める際、2つのパケットを用いている場合($m = 2$)しか想定していない。ゆえに、DoS攻撃に対する、Kolmogorov Complexityの有効性を調べるためには、(2)式における、mの値も変化させて、実験を行う予定である。これは、どのmの値において、Kolmogorov ComplexityがDoS検知に最も有効であるのか知る必要があるからである。

また、Syn floodに限らず他の攻撃においても、どの程度Kolmogorov Complexityが有効であるかについて調べる必要がある。同様に、正規の通信も、今回以外の方法を試みて得られる結果の変化について考察を行う予定である。

謝辞

本研究の一部は、平成15年度文部科学省科学研究費補助金(研究課題番号: 特定領域研究(2) 15017226)の援助を受けた。

参考文献

[1] "Distributed Denial of Service (DDoS) Attacks/tools". http://staff.washington.edu/dittrich/misc/ddos/

[2] Felix Lau, Stuart H. Rubin, Michael H. Smith, and Ljiljana Trajovic, "Distributed Denial of Service Attacks", In Proc of IEEE International Conference on Systems, Man, and Cybernetics, pp.2275–2280, October 2000.

[3] Jussipekka Leiwo, Tuomas Aura, and Pekka Nikander: "Towards Network Denial of Service Resistant Protocols", In Proceedings of the 15th International Information Security Conference (IFIP/SEC 2000), Kluwer, pp.301–310, August 2000.

[4] K. Matsuura and H. Imai: "Modified Aggressive Modes of Internet Key Exchange Resistant against Denial-of-Service Attacks", IEICE Transactions on Information and Systems, vol.E83-D, no.5, pp.972–979, May 2000.

[5] Hassan Alifri: "IP Traceback:A New Denial-of-Service Deterrent?" (IEEE SECURITY & PRIVACY MAY/JUNE 2003 pp.24-31)

[6] P.Ferguson, D.Senie: "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing", RFC 2827, May 2000.

[7] D. Denning, "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, vol.13, no.2, pp.222-232, 1987.

[8] M. Thottan and C. Ji, "Proactive Anomaly Detection Using Disributed Intelligent Agents", IEEE Network, vol.12, no.5, pp.21-27, 1998.

[9] R. A. Kemmerer and G. Vigna, "Intrusion Detection: A Brief History and Overview", Supplement to IEEE Computer, Security & Privacy, pp.27-30, 2002.

[10] Jelena Mirkovic et al.:"A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms" , Tech. Report, UCLA CSD, CSD-TR-020018, 2002

[11] S. Cheung and K. N. Levitt, "Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection", In Proc. of New Security Paradigms Workshop '97, pp.94-106, September 1997.

[12] J. Sun, H. Jin, H. Chen, Q. Zhang, and Z. Han: "A Compound Intrusion Detection Model". In Proc. of ICICS (5th International Conference on Information and Communication's Security), LNCS 2836, pp.370–381, October 2003.

[13] A.B.Kulkarni, S.F.Bush, S.C.Evans: "Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics". Tech. Report, GE Research & Development Center, 2001CRD176, December 2001 (Class 1)

[14] T.Cover, J.Thomas:"Elements of Information Theory". John Wiley & Sons, Inc., New York, pp.144-153, 1991

[15] Ming Li, Paul Vitanyi:"An Introduction to Kolmogorov Complexity and Its Applications". Springer, Berlin, 1993

[16] S.C.Evans et al.:"Kolmogorov Complexity Estimation and Analysis".Tech. Report, GE Research & Development Center, 2002GRC177, October 2002(Class 1)