Algorithms and Tools for Anonymization of the Internet Traffic

Tanjila Farah
tfarah@sfu.ca

Communication Networks Laboratory
School of Engineering Science
Simon Fraser University
Roadmap

- Introduction
- Collection of network traffic
- Anonymization fields, algorithms, and tools
- Anonym tool
- Conclusion, future work, and references
Roadmap

- Introduction
- Collection of network traffic
- Anonymization fields, algorithms, and tools
- Anonym tool
- Conclusion, future work, and references
Internet is the easiest and the fastest medium for communication

Measurement, characterization, and classification of Internet traces help network security

Real-time network analysis relies on collection of trace logs

Sharing traces may reveal the network architecture, user identity, and user information
Anonymization

- Modifies network traces to protect user identity
- Removes the ability to identify the connection between two end-users
- Preserves the usefulness of the datasets
- Considers the type of analysis that may be performed
- Considers the requirements of the company sharing the datasets
Contributions

- Developed code in gawk to parse pcap and mrt input files
- Anonym tool:
  - developed the tool
  - developed the IPv6 address anonymization technique
  - implemented data analysis and visualization options
  - validated the tool performance
Roadmap

- Introduction
- Collection of network traffic
- Anonymization fields, algorithms, and tools
- Anonym tool
- Conclusion, future work, and references
Collection of network traffic

- Internet is a collection of ASes exchanging information and delivering data
- Process of delivering data creates network traffic
- Network performance and QoS rely on network traffic characteristics
- Analyzing and understanding the network traffic helps ensure network security and QoS
- Network traffic collection helps:
  - traffic engineering
  - discovering the Internet topology
  - analyzing network security

AS : Autonomous System
QoS: Quality of Service
Role of traffic engineering

- **Network troubleshooting:**
  - deals with issues that *disrupt or degrade the performance of a network*: incorrect network address assignments and network anomalies

- **Protocol debugging:**
  - analyzes the *existing and new protocols and performance of applications* to determine required improvements

- **Workload characterization:**
  - examines the *growth of network traffic volume* due to new applications, protocols, and increasing number of users
Role of traffic engineering (cont.)

- **Network performance evaluation:**
  - estimates the network QoS by measuring traffic throughput and response time

- **Capacity planning:**
  - deals with network planning and managing by measuring bandwidth usage and availability
Discovering the Internet topology is important for:

- simulating deployed networks
- managing networks
- mapping a network to determine location of the nearest servers and ISPs
- designing and implementing new topology-aware protocols and algorithms
Network security analysis

- Monitors policies adopted by network administrators to prevent the intruders from misusing the network.

- It encompasses:
  - determining abnormal events: anomalies, attacks, and viruses
  - testing network firewalls
  - controlling access and network usage
Network trace collection

- **BCNET:**
  - British Columbia's advance communication network
  - collected data are *private* and are only shared with the **CNL**
  - data are collected in the *pcap* format

- **Cooperative Association for Internet Data Analysis (CAIDA):**
  - collects, monitors, and visualizes various Internet data
  - collected data are *public*
  - data are collected in *pcap* and *text* formats
Network trace collection

- **Route Views:**
  - project at the University of Oregon
  - provides data and tools to the network administrators
  - collected data are *public*
  - data are collected in the *mrt* format

- **Réseaux IP Européens (RIPE):**
  - supports network operators in Europe, Middle East, Asia, and Africa
  - collected data are *public*
  - data are collected in the *mrt* format
Roadmap

- Introduction
- Collection of network traffic
- Anonymization fields, algorithms, and tools
- Anonym tool
- Conclusion, future work, and references
Anonymization fields

- Network traffic logs include data **packet headers**, which contain various fields:
  - time-stamp
  - IP addresses
  - MAC addresses
  - packet length
  - protocol

Anonymization algorithms

- **Black marker:**
  - *deletes* all the information or *replaces* the information by a fixed value

<table>
<thead>
<tr>
<th>Time</th>
<th>IP</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0534</td>
<td>253.36.88.92</td>
<td>143</td>
</tr>
</tbody>
</table>

- **Enumeration:**
  - *sorts* the dataset, *chooses* a value higher than the first value, and *adds* the value to all data points

<table>
<thead>
<tr>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>143</td>
</tr>
<tr>
<td>60</td>
</tr>
<tr>
<td>1514</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>203</td>
</tr>
<tr>
<td>120</td>
</tr>
<tr>
<td>1574</td>
</tr>
</tbody>
</table>
Anonymization algorithms (cont.)

- **Precision degradation:**
  - removes the most *precise components* of a data field

<table>
<thead>
<tr>
<th>IP un-anonymized</th>
<th>IP anonymized</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.017851</td>
<td>1.017000</td>
</tr>
<tr>
<td>1.017852</td>
<td>1.017000</td>
</tr>
<tr>
<td>1.017915</td>
<td>1.017000</td>
</tr>
</tbody>
</table>

- **Prefix-preserving:**
  - if two IP addresses share the first $n$ bits then their anonymized IP addresses will also share the first $n$ bits

<table>
<thead>
<tr>
<th>IP un-anonymized</th>
<th>IP anonymized</th>
</tr>
</thead>
<tbody>
<tr>
<td>112.116.186.8</td>
<td>235.251.46.4</td>
</tr>
<tr>
<td>115.23.40.51</td>
<td>240.48.153.85</td>
</tr>
<tr>
<td>112.116.186.8</td>
<td>235.251.46.4</td>
</tr>
<tr>
<td>115.23.40.51</td>
<td>240.48.153.85</td>
</tr>
</tbody>
</table>
Anonymization algorithms (cont.)

- **Random shift:**
  - shifts each data point by adding a random number

<table>
<thead>
<tr>
<th>Packet length un-anonymized</th>
<th>Packet length anonymized</th>
</tr>
</thead>
<tbody>
<tr>
<td>143</td>
<td>150</td>
</tr>
<tr>
<td>60</td>
<td>230</td>
</tr>
<tr>
<td>1514</td>
<td>1674</td>
</tr>
</tbody>
</table>

- **Truncation:**
  - deletes the n least significant bits from an IP or MAC address

<table>
<thead>
<tr>
<th>MAC address</th>
<th>Anonymized MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco_e7:a1:c0 (00:1b:0d:e7:a1:c0)</td>
<td>Cisco_0:0:0 (00:1b:0d:0:0:0)</td>
</tr>
<tr>
<td>JuniperN_3e:ba:bd (78:19:f7:3e:ba:bd)</td>
<td>JuniperN_0:0:0 (78:19:f7:0:0:0)</td>
</tr>
</tbody>
</table>
Anonymization algorithms (cont.)

- **Reverse truncation:**
  - deletes the $n$ most significant bits from an IP or MAC address

<table>
<thead>
<tr>
<th>MAC address</th>
<th>Anonymized MAC address</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cisco_e7:a1:c0</strong> (00:1b:0d:e7:a1:c0)</td>
<td><strong>Cisco_e7:a1:c0</strong> (0:0:0:e7:a1:c0)</td>
</tr>
<tr>
<td><strong>JuniperN_3e:ba:bd</strong> (78:19:f7:3e:ba:bd)</td>
<td><strong>JuniperN_3e:ba:bd</strong> (0:0:0:3e:ba:bd)</td>
</tr>
</tbody>
</table>
Anonymization tools

- Cryptography based Prefix-preserving Anonymization: Crypto-PAn
- Anontool
- Framework for Log Anonymization and Information Management: FLAIM
Crypto-PAn

Properties of Crypto-PAn:
- one-to-one mapping
- prefix-preserving anonymization
- consistent across traces
- cryptography-based

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>IP address</td>
</tr>
<tr>
<td>0.000010</td>
<td>10.1.3.143</td>
</tr>
<tr>
<td>0.000015</td>
<td>10.1.3.156</td>
</tr>
<tr>
<td>Time</td>
<td>IP address</td>
</tr>
<tr>
<td>0.000010</td>
<td>117.14.240.136</td>
</tr>
<tr>
<td>0.000015</td>
<td>117.14.240.85</td>
</tr>
</tbody>
</table>
Anontool supports per-field anonymization.

Supports log files: pcap, netflow v5, and netflow v9.

Four-step anonymization process:

- **cooking function**
  - assembles the flows according to protocols

- **filtering function**
  - distinguishes the flows according to protocol and determine policy for anonymization

- **anonymization function**
  - anonymizes the fields according to policy

- **un-cooking function**
  - re-assembles the flows in the original format
FLAIM

- Supports an XML based policy
- Parsing modules are written based on the XML policy
- Supports log files: pcap, iptable, nfdump, and pacct
- FLAIM architecture consists of a module and a core:
  - the module provides policies to identify type of the log file
  - the core loads libraries responsible for anonymization
Roadmap

- Introduction
- Collection of network traffic
- Anonymization fields, algorithms, and tools
- Anonym tool
- Conclusion, future work, and references
Anonym tool: functions

- Parses pcap and mrt files
- Anonymization options:
  - black marker, prefix-preserving, reverse-truncation, precision degradation, random shift, and truncation
- Data analysis options:
  - volume (bytes), volume (packets), volume curve fitting, throughput, empirical distribution, packet length distribution, protocol distribution, boxplot, and PDF and CDF curve fitting
Anonym tool: functions

- Options for the K-S test:
  - determines if a dataset matches a tested distribution
  - provide options to test: normal, gamma, Weibull, exponential, Rayleigh, and lognormal distributions

- Additional options:
  - display anonymization results and analysis graphs
  - clear and upload new file
  - save figures and anonymization results

K-S: Kolmogorov-Smirnov

Data analysis options

- We analyzed the effect of anonymization on the dataset by using the analysis options implemented in the Anonym tool.
Data analysis option: volume

- Number of **bits or packets** per second
- Identifies the **pattern of traffic flow** through a network
- Shown are BGP, TCP, and UDP traffic volume:

![Graphs](image)

**TCP**: Transmission Control Protocol  
**BGP**: Border Gateway Protocol  
**UDP**: User Datagram Protocol
Data analysis option: volume

Statistics of packet length:
- an enumeration algorithm is applied to the dataset

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Un-anonymized dataset (bits)</th>
<th>Anonymized dataset (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>60</td>
<td>160</td>
</tr>
<tr>
<td>Maximum</td>
<td>1,514</td>
<td>1,614</td>
</tr>
<tr>
<td>Mean</td>
<td>246.2475</td>
<td>346.2475</td>
</tr>
<tr>
<td>Median</td>
<td>157</td>
<td>257</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>259.4509</td>
<td>259.4509</td>
</tr>
</tbody>
</table>
Data analysis option: volume curve fitting

- **Run-sequence**: displays a graphical representation of a dataset
- **Fitting curves** to a dataset: Fourier, Gaussian, Weibull, exponential, polynomial, and sum of sine distributions
Data analysis option: protocol distribution

- Provides an overview of various protocols occupancy in the network
- Classifies IP, UDP, TCP, ICMP, DNS, and BGP traffic

ICMP: Internet Control Message Protocol
DNS: Domain Name Service
Data analysis option: packet length distribution

- Displays the histogram plot of a dataset
- Indicates appropriate distribution model of a dataset
- Significant percentage of packets are 150 bytes for un-anonymized dataset and 250 bytes for anonymized dataset as shown:

Data analysis option: fitting PDFs and CDFs

- PDF and CDF indicate the probability that the structure of a dataset follow certain distribution
- Provides options to fit thirteen distributions to PDF and CDF distribution curves of a dataset

Anonym tool: GUI

- Graphical user interface

Operational diagram

- Prefix-preserving option
Functions: code

Call function for separating IPv4 and IPv6 flows

```matlab
iporder=[];
f=fopen(name);
f4=fopen(name4,'w');
f6=fopen(name6,'w');
while 1 % For each line
    line = fgetl(f);
    if (strfind(line, ':'))
        iporder=[iporder 1];
        fprintf(f6, '%s\n', line);
    else
        iporder=[iporder 0];
        fprintf(f4, '%s\n', line);
    endif (feof(f))
    break;
end
fclose(f);
close(f4);
close(f6);
[time4,IPv4source,IPv4destination,protocol4,ipv4pktlength] = ipv4decode(name4);
[time6,IPv6s,IPv6d,protocol6,pktlength] = ipv6decode(name6);
```

Call prefix-reserving-function to anonymize IPv4

```matlab
v4source = 'v4SColumn.txt';
% Output of decoded IPv4 destination column
v4destination='v4DColumn.txt';
% This gives the size of the input file
inputsize= size(IPv4source,1);
% Time in defind zero because at this point we are not decoding the
time data. This need to be fixed.
% time=0;
% Writing the output of IPv4 decode in a files v4SColumn.txt and
v4DColumn.txt.
% Source and destination is writen in two files because Crypto-PAN
takes
% input as it this formate (time length, address)
ipv4writefile (v4source,v4destination, time4, ipv4pktlength,IPv4source,IPv4destination,inputsize);
v4Sanonymized='v4sourceanonymized.txt';  % IPv4 source address anonymized
v4Danonymized='v4destinationanonymized.txt';   % IPv4 destination address anonymized
[ss]=dos(['cryto_run.exe ' v4source ' > ' v4Sanonymized]);
[ss]=dos(['cryto_run.exe ' v4destination ' > ' v4Danonymized]);
```
Call prefix-preserving-function to anonymize IPv6

IPv4s=IPv6s(:,1:4);
IPv4d=IPv6d(:,1:4);
lines = size(IPv6s,1);
namenv4s='10outv4s.txt';
namenv4d='10outv4d.txt';
time=0;
ipv6toipv4writefile(namenv4s,namenv4d, time6, pktlength, IPv4s, IPv4d,lines);
namenv4sout='10outv4sout.txt';
namenv4dout='10outv4dout.txt';
[s,s]=dos(['cryto_run.exe ' namenv4s ' > ' namenv4sout]);
[s,s]=dos(['cryto_run.exe ' namenv4d ' > ' namenv4dout]);

[IPv4sout,IPv4dout, timeout, pktlengthout]=ipv4toipv6readfile(namenv4sout, namenv4dout, lines);
Anonymized IPv6 address output
nameano='10outv6ano.txt';
writeipv6anon(time6,IPv6s,IPv6d,IPv4sout,IPv4dout,protocol6,pktlengthout,nameano);

Call precision-degradation-function to anonymize IPv4 flow time-stamps

[IPv4source,IPv4destination, time4, ipv4pktlength]=ipv4readfile(v4Sanonymized, v4Danonymized, inputsize);

precision degradation timeanony4=fix(time4*100)/100;
nameano='10outtimev4ano.txt';
writeipv4anon(timeanony4,IPv4source,IPv4destination,protocol4,ipv4 pktlength,nameano)

[IPv4sout,IPv4dout, timeout, pktlengthout]=ipv4toipv6readfile(namenv4sout, namenv4dout, lines);
precision degradation

Call precision-degradation-function to anonymize IPv6 flow time-stamps

[IPv4sout,IPv4dout, timeout, pktlengthout]=ipv4toipv6readfile(namenv4sout, namenv4dout, lines);
Anonymized IPv6 address output
nameano='10outv6ano.txt';
writeipv6anon(time6,IPv6s,IPv6d,IPv4sout,IPv4dout,protocol6,pktlengthout,nameano);
Implementation of the Anonym tool was validated using various tests:

<table>
<thead>
<tr>
<th>Fields</th>
<th>Anonym</th>
<th>Anontool</th>
<th>FLAIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source 64.251.87.209</td>
<td>0.29.105.18</td>
<td>110.13.240.136</td>
<td>103.51.250.0</td>
</tr>
<tr>
<td>Destination 64.251.87.210</td>
<td>0.29.105.17</td>
<td>110.13.246.137</td>
<td>103.51.250.28</td>
</tr>
</tbody>
</table>
The **Anonym tool**: results

- **Per-field anonymization results:**

<table>
<thead>
<tr>
<th>Time-stamp</th>
<th>IPv4 and IPv6</th>
<th>Packet length</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td><strong>Un-anonymized dataset</strong></td>
<td></td>
</tr>
<tr>
<td>0.000000</td>
<td>2001:4958:10:2::2</td>
<td>2001:4958:10:2::3</td>
</tr>
<tr>
<td>2.410144</td>
<td>64.251.87.209</td>
<td>64.251.87.210</td>
</tr>
<tr>
<td>4.563551</td>
<td>206.47.102.206</td>
<td>206.47.102.201</td>
</tr>
<tr>
<td>0.000000</td>
<td><strong>Anonymized dataset</strong></td>
<td></td>
</tr>
<tr>
<td>0.000000</td>
<td>8:A7:10:2:0:0:2</td>
<td>8:A7:10:2:0:0:0:3</td>
</tr>
<tr>
<td>1.170000</td>
<td>8:A7:10:2:0:0:2</td>
<td>8:A7:10:2:0:0:0:3</td>
</tr>
<tr>
<td>2.410000</td>
<td>0.29.105.18</td>
<td>0.29.105.17</td>
</tr>
<tr>
<td>4.560000</td>
<td>240.48.153.6</td>
<td>240.48.153.0</td>
</tr>
</tbody>
</table>
Roadmap

- Introduction
- Collection of network traffic
- Anonymization fields, algorithms, and tools
- Anonym tool
- Conclusion, future work, and references
Conclusions

- The Anonym tool provides options to anonymize time, IPv4 and IPv6 addresses, MAC addresses, and packet length data.
- Supports log files in mrt and pcap formats.
- Provides options to analyze the datasets.
- Provides options to apply the K-S test on the datasets.
- Analysis of un-anonymized and anonymized datasets indicates insignificant variations.
Future work

- The **Anonym** tool may be enhanced to support other log formats: netflow, iptable, and pccat
- **Additional** anonymization algorithms may be implemented: binning, hash, partitioning, permutation, and random noise addition
- Anonymization of **additional fields** may be implemented: port numbers, TCP window size, and IP ID number
References


References


Acknowledgements

- Prof. Veselin Jungic, Chair
- Prof. Parvaneh Saeedi, Supervisor
- Prof. Emeritus Stephen Hardy, Examiner
- Prof. Ljiljana Trajković, Senior Supervisor
Acknowledgements

- Toby Wong, BCNET
- CNL mates
- Eva María Cavero Racaj, Universidad de Zaragoza
Thank You!