Practical Data Hiding in TCP/IP

Kamran Ahsan
and
Deepa Kundur
Bell Canada Junior Chair-holder in Multimedia

Electrical and Computer Engineering
University of Toronto
Agenda

- Introduction
- Problem Formulation
- Previous Work
- Proposed Techniques
- Application Scenarios
- Conclusions
Introduction

- “Open” specifications of the Internet
  - Communications
  - Connectedness
  - Collaboration

- Security in the Internet an afterthought
What is this Paper About?

- Can we identify practical covert channels in TCP/IP?
- How can these channels be used to enhance network processing and security?

Deepa Kundur, University of Toronto

ACM Multimedia and Security Workshop
Our Focus

- Covert channels in computer networks
- Data hiding through network packet streams
- Network behavior on packets carrying covert data
- Associated Applications
Covert Channels

- Channel used, but not designed for info transmission
  - Can violate security policy
  - Shared resources, redundancies, multiple interpretations
  - Storage and Timing Channels
Data Hiding (DH)

- Methodology by which to exploit the presence of covert channels
  - Cover object + Covert Data = Stego-object

- Existing research focused on digital images as cover object

We use network packet streams as the cover object
Deepa Kundur, University of Toronto

Framework

- Covert channel piggy-backed on legitimate overt channel
  - Stego Algorithm should not affect overt channel
  - Covert data undetectable by network filters
Previous Work

Covert Channel Based

2. Wolf (1989): LAN protocols
4. Rowland (1997): TCP/IP; proof of the concept
5. Ackermann et al. (2000):
   - Weakening of layered concept
   - Additional info. in network packets

Networks Based
The Complete Picture

1. Capacity of covert comm.

3. OSI
4. TCP/IP

SENDERSender

Embedding

1. LAN
2. LAN - Protocols

Detecting

3. OSI
4. TCP/IP

RECEIVERReceiver

5. Application Scenarios

Techniques of DH in TCP/IP + (Embedding & Extraction Scenarios) + Application Scenarios + Effects of Covert Communication on Overt Channels + Simulation and Testing
Proposed Algorithms

- **Illustrative Examples**
  - Packet header manipulation
  - Packet “sorting”

- Make use of chaotic mixing
Packet Header Manipulation

- IPv4
- Analyzed protocol header
  - Looking for redundancies
  - Multiple interpretations of features and policies
- Develop scenarios wrt network environment
DH Scenario 1

- Multiple interpretation of fragmentation strategy
- Utilize flags field; DF (Do not Fragment) bit

<table>
<thead>
<tr>
<th>Datagram</th>
<th>16-bit Ident. field</th>
<th>3-bit flag field</th>
<th>13-bit frag. offset</th>
<th>16-bit total length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XX...XX</td>
<td>010</td>
<td>00...00</td>
<td>472</td>
</tr>
</tbody>
</table>

Covertly Communicating ‘1’

<table>
<thead>
<tr>
<th>Datagram</th>
<th>16-bit Ident. field</th>
<th>3-bit flag field</th>
<th>13-bit frag. offset</th>
<th>16-bit total length</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>XX...XX</td>
<td>000</td>
<td>00...00</td>
<td>472</td>
</tr>
</tbody>
</table>

Covertly Communicating ‘0’
DH Scenario 2

- Make use of Sequence Number field
- Must be “unique” for a given source-destination pair
Toral Automorphisms (TAs)

- Chaotic systems
- Watermarking in digital images
- Toral automorphism matrix

\[ A = \begin{bmatrix} 1 & 1 \\ k & k + 1 \end{bmatrix} \]

maps all points on a lattice

\[ \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ k & k + 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \pmod{K} \]

periodic; recurrence time, \( P \)
Toral Automorphisms (2)

Generation of sequence numbers:

- **Main key** = Size of the Lattice; $K$
- **Sub key** = Parameter of TA matrix; $k$
- **Third key** = No. of TA applications

TAs provides structured scrambling and enables Alice and Bob to communicate covertly.
DH Scenario 2

- **Alice’s End**
  1. Selection of keys
  2. Formation of a look-up table; sorted sequence matched with alphabet
  3. Conversion to binary
  4. **Appending** randomly generated 8 bits to form 16-bit Identification field

- **Bob’s End**
  - Generation of the look-up table
  - Deciphering of the Identification field thereafter
### DH Scenario 2

**Generation of Identifier by chaotic mixing**

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Alphabets</th>
<th>Seq.for 8th iter</th>
<th>Binary Equ.</th>
<th>Encoded in 4-bit</th>
<th>Ident. Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
<td>0000 00001</td>
<td>0 1</td>
<td>0 1 X X</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>14</td>
<td>0000 1110</td>
<td>0 E</td>
<td>0 E X X</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>9</td>
<td>0000 1001</td>
<td>0 9</td>
<td>0 9 X X</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>22</td>
<td>0001 0110</td>
<td>16</td>
<td>1 6 X X</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>4</td>
<td>0000 0100</td>
<td>0 4</td>
<td>0 4 X X</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>17</td>
<td>0001 0001</td>
<td>1 1</td>
<td>1 1 X X</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>25</td>
<td>0001 1001</td>
<td>1 9</td>
<td>1 9 X X</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>12</td>
<td>0000 1100</td>
<td>0 C</td>
<td>0 C X X</td>
</tr>
<tr>
<td>24</td>
<td>X</td>
<td>24</td>
<td>0001 1000</td>
<td>1 8</td>
<td>1 8 X X</td>
</tr>
<tr>
<td>25</td>
<td>Y</td>
<td>6</td>
<td>0000 0110</td>
<td>0 6</td>
<td>0 6 X X</td>
</tr>
<tr>
<td>26</td>
<td>Z</td>
<td>19</td>
<td>0001 0011</td>
<td>1 3</td>
<td>1 3 X X</td>
</tr>
</tbody>
</table>
Potential Applications

- Enhanced filtering criteria in firewalls
- Security tied to the content – client-server architecture
- Content delivery networks
Data Hiding by Packet Sorting

- Sorting: \( n \) objects can store \( \log_2(n!) \) bits

- Packet “sorting” / “resorting” at network layer
  - Reference = Sequence number field of IPSec
  - No major modification in header fields

- Sorting: chaotic mixing
- Resorting: best sequence estimation
Sorting / Resorting Process

- Two keys:
  - Main key = $K$
  - Sub key = $k$

- Covert data:
  - third key = seq. no.

- From:
  - Main key = $K$
  - Sub key = $k$

Bob *estimates* the third key, the covert message
Best Sequence Estimation

- Out of order delivery by the Internet layer
  - Out of orderedness is prevalent and asymmetric
  - Introduction of packet position errors
  - Small scale reordering; Paxson and Mogul findings

- Longest Subsequence (LSS) Method
The Technique

- Received Sequence, $R_x^r$
- One of the Sent Sequences, $S(i)_x$

Finding # of position errors (PE)
- Identifying zeros; similar position packets
- Subtracting # of zeros from main key “K”, PE is obtained

If PE <= threshold PE

YES

NO

Impossible Sequence

The Right Shift Absolute Subtraction Method (RSAS)
1. From the subtraction process, identify zeros; similar-position packets
2. Truncate the last packet of the sent sequence and the first packet of the received sequence
3. Repeat 2 till the first packet of the sent sequence undergoes RSAS with the last packet of the received sequence (i.e. “K” steps)
   a. For each one of the steps, identify zeros.
4. A resultant sub-sequence is achieved from positions in the sequences where zeros are identified;
5. Count the total # of zeros

Deepa Kundur, University of Toronto

ACM Multimedia and Security Workshop
The Technique (2)

If the resultant sub-sequence *conforms* to both the sequences

**The Right Shift Absolute Subtraction Method (RSAS)**

To a point where, # of zeros = (total # of zeros) – 1:

1. Repeat steps 1 to 5
2. Result would be the sub sequence having elements (packets) one less than the sub sequence achieved in the previous step

If the resultant sub-sequence *conforms* to both the sequences

LSS based Best Sequence

NO

Error Sequence

YES

LSS based Best Sequence
Simulation and Testing

- Process is simulated for $K=4, \ldots, 8$ and $k=1$
- A practical communication network; introducing 3 to 6 position errors in packet sequence.
Position Error (PE) Scenarios

- Small scale position errors (Paxson & Mogul)
- For specific packet sequence: Consider
  - All position errors
  - All permutations equally likely
- Evident sequences or/and LSS based best estimate sequences are highly desirable
### Analysis - PE Scenarios

Which sent sequence is most likely to be mapped at Bob’s end?

<table>
<thead>
<tr>
<th>Received Sequence Category</th>
<th>Seq.1 S(1)</th>
<th>Seq.2 S(2)</th>
<th>Seq.3 S(3)</th>
<th>Seq.4 S(4)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impossible</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>541</td>
</tr>
<tr>
<td>Error</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>Evident</td>
<td>33</td>
<td>36</td>
<td>40</td>
<td>34</td>
<td>143</td>
</tr>
<tr>
<td>Best Estimate</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>38</strong></td>
<td><strong>41</strong></td>
<td><strong>44</strong></td>
<td><strong>41</strong></td>
<td><strong>720</strong></td>
</tr>
</tbody>
</table>

Main key= 6(imprvd.); Sent seq.= 4; Network behavior: 3 PE or less
Usage Scenarios

- Packet Sorting/Resorting
  - Preliminary authentication in IPSec
  - Enhanced anti-traffic analysis
  - Enhanced security mechanisms for IPSec protocols
Conclusion

Covert channels in networks – a step forward

Network processing and security can be reinforced by integrating steganography with existing security architecture

- Packet header manipulation – network processing and security services
- Packet sorting – network security mechanisms
**Q&A**

**IPSec**

\[ A = \begin{bmatrix} 1 & 1 \\ k & k + 1 \end{bmatrix} \]

<table>
<thead>
<tr>
<th>4-bit Ver</th>
<th>4-bit IHL</th>
<th>8-bit TOS</th>
<th>16-bit Tot. Len.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>0101</td>
<td>XXXXXXXX</td>
<td>XXXXXXXXXXXXXXX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16-bit Idnt.</th>
<th>3-bit Flags</th>
<th>13-bit Frag Off.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0100</td>
<td>XXX</td>
<td>XXXXXXXXXXXXXXX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8-bit TTL</th>
<th>8-bit Protocol</th>
<th>16-bit Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXXXXX</td>
<td>XXX</td>
<td>XXXXXXXXXXXXXXX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>32-bit Source Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>32-bit Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
</tbody>
</table>