NETWORK COVERT TIMING CHANNELS

Confidential Data

Shared Code Book

Sender

Receiver

Packet Transmission Times

Packet Reception Times

Confidential Data
**Recent Work**

- **IP Covert Timing Channels: Design and Detection, CCS’04**
  by S. Cabuk, C. Brodley, and C. Shields
  - data rate 16.67 bits/sec (error rate 2%)

- **Keyboards and Covert Channels, USENIX Security’06**
  by G. Shah, A. Molina, and M. Blaze
  - low data rate

- **Capacity Bounds for BSTC, ISIT ’07**
  by S. Sellke, C. C. Wang, N. Shroff, and S. Bagchi
  - Information Theoretical Analysis
**Our Contribution**

- Design of **two** Timing Channels:
  - **Timing Channel 1** – achieves higher leak rate:
    - significantly improved data rate (5 x)
  - **Timing Channel 2** - concealable:
    - mimics i.i.d. normal traffic
    - computationally indistinguishable from i.i.d. normal traffic

- Validation of the design
  - Software implementations
  - Experiments on PlanetLab nodes
OUTLINE

- Design of High Rate Timing Channel
- Experimental Results
- Concealable Timing Channels
**Network Timing Channel Design**

- **L-bits to n-packets scheme:**
  - Maps L-bits to n-packets inter-transmission times

- **Two design parameters:** \( \Delta \) and \( \delta \)
  - A 4-bits to 2-packets scheme (\( \Delta = 60 \) ms, \( \delta = 10 \) ms)
  - \( T_1, T_2 \): packet inter transmission times

<table>
<thead>
<tr>
<th>Bit String</th>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T1, T2)</td>
<td>(60,60)</td>
<td>(60,70)</td>
<td>(70,60)</td>
<td>(70,70)</td>
<td>(60,80)</td>
<td>(100,100)</td>
</tr>
</tbody>
</table>

- \( T_1, T_2, T_3, \ldots, T_n \) takes values from the set
  \[ E = \{ T: T = \Delta + k \times \delta, \ k=0, 1, 2, \ldots \} \]
EXAMPLE OF DECODING ERROR

- Decoding error caused by small $\delta = 8$ ms
- Transmission delays: 30ms +/- 5ms

code #1 maps 00 to $T1=60$ ms

code #2 maps 10 to $T1=68$ ms
**DESIGN CHALLENGE**

- Determine the optimal values of L and n
- Two simple examples ($\Delta = \text{60 ms}$, $\delta = \text{20 ms}$):
  - 2-bits to 1-packets scheme: 22 bits/sec
  - 4-bits to 1-packets scheme: 19 bits/sec

<table>
<thead>
<tr>
<th>Bit strings</th>
<th>00</th>
<th>10</th>
<th>01</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>60</td>
<td>80</td>
<td>100</td>
<td>120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit strings</th>
<th>0000</th>
<th>1001</th>
<th>...</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>60</td>
<td>80</td>
<td>...</td>
<td>360</td>
</tr>
</tbody>
</table>
DATA RATE FOR TYPE 1 TIMING CHANNEL

- **K**: an auxiliary parameter
  - Used to bound the packet transmission time

- *(n, K)-code*: a special L-bits to n-packet code
  - \( T(i) = \Delta + k(i) \cdot \delta \)
  - \( K: \sum k(1)+k(2)+\ldots+k(n) \leq K \)
  - total transmission time \( \leq n \Delta + K \delta \)

- **Fact**: \( 2^L \leq C(n+K, K) \);
  - choose \( L = \text{floor}(\log_2 C(n+K, K)) \)
DATA RATE FOR TYPE 1 TIMING CHANNEL

○ Lemma: Given the system parameters \((\Delta, \delta)\), the data rate \(R(n,K)\) of an \((n, K)\)-code

\[
R(n, K) \approx \frac{\log_2 C(n + K, K)}{n \cdot \Delta + \frac{n}{n+1} \cdot K \cdot \delta} \quad \text{bits/sec.}
\]

○ Main Result:
  ○ Optimal Data Rate \(R^*(n)\) given \((\Delta, \delta)\):

\[
R^*(n) \approx \max_{K \geq 0} \frac{\log_2 C(n + K, K)}{n \cdot \Delta + \frac{n}{n+1} \cdot K \cdot \delta} \quad \text{bits/sec.}
\]
PLOT OF DATA RATE $R(n,K)$

- $\Delta = 50$ ms, $\delta = 10$ ms
  - $n = 3$
    - $R^*(3) = 37$ b/s
    - $L^* = 9$
    - 9-bits to 3-packets
  - $n = 5$
    - $R^*(5) = 38$ b/s
    - $L^* = 15$
    - 15-bits to 5-packets

- Performance Tradeoffs
  - $R^* = 39$ b/s requires 66-bits to 32-packets scheme
OUTLINE

- Design of Timing Channel 1
- Experimental Results
- Concealable Timing Channels
EXPERIMENTS
DECODING ERRORS

Princeton and Purdue

Day 1  Day 2  Day 3  Day 4  Day 5  Day 6  Day 7  Day 8  Day 9  Day 10

Decoding Error Rate

(50,10) (50,5) (40,10) (40,5) (30,10) (30,5) (20,10) (20,5) (10,10) (10,5)

(current result (CCS’04):
data rate: 17 b/s
error rate: 2%)
ERROR CORRECTION

- Net error-free rate = raw rate * \((1 - H_{255}(\text{byte error rate})/8)\)
  - 8% error \(\Rightarrow\) 87% raw data rate
  - 4% error \(\Rightarrow\) 93%
  - 2% error \(\Rightarrow\) 96%
  - 1% error \(\Rightarrow\) 98%
DECODING ERRORS

Princeton and Purdue

Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Day 10
--- | --- | --- | --- | --- | --- | --- | --- | --- | ---
36 | 37 | 37 | 43 | 44 | 51 | 55 | 63 | 60 | 73
42 | 51 | 49 | 63 | 73 | 76 | 108 | 82 | 124 | (50,10) (50,5) (40,10) (40,5) (30,10) (30,5) (20,10) (20,5) (10,10) (10,5)
OUTLINE

- Design of Timing Channel 1
- Experimental Results
- Concealable Timing Channel
TYPE 2 TIMING CHANNEL: CONCEALABLE

Goal:
- Immune against current and future detection

How do we achieve this goal?
- Mimic the statistical property of i.i.d. normal traffic
- Computationally indistinguishable from i.i.d. normal traffic

Timing channel is a serious security concern
**CONCEALABLE TIMING CHANNEL**

**Achieving Design Goals:**
- Mimics statistical property
- Computationally indistinguishable from i.i.d. normal traffic

### 1: Codeword Look Up.

\[ c(i) \rightarrow (x(2i-1), x(2i)) = (15/16, 3/16) \]

### 2: Codeword Masking using C SPRNG.

a) C SPRNG \(\rightarrow u(1), u(2), \ldots u(2n)\)

b) \(r(i) = x(i) + u(i) \mod 1\), for \(i = 1, \ldots, 2n\)

### 3: Inter-Transmission Time Generation:

\[ T(i) = F^{-1}(r(i)) \]

\(F(x): CDF\) of a normal traffic

---

**Decoding:**
- Reversal of the above three steps
**CONCEALABLE TIMING CHANNEL**

- **Advantages:**
  - Immune from current and future detection
  - Same codebook for different traffic patterns
  - No handshaking necessary

- **Experiments:**
  - Purdue ➔ Princeton Telnet (i.i.d. Pareto)
  - Data rate: 5 bits/sec
  - Error rate: 1%
CONCLUSION

- Demonstrated considerably higher threat of information leaking through the network covert timing channels
  - leaks information at much higher rate
  - hard to detect
    - leaking information long term at constant rate (e.g. 5 b/s)

- Future Direction:
  - Efficient algorithm to mimic correlated traffic, such as HTTP traffic
Thank You!
DECODING ERRORS

Purdue and Zurich

Day 1  Day 2  Day 3  Day 4  Day 5  Day 6  Day 7  Day 8  Day 9  Day 10

<table>
<thead>
<tr>
<th>(Delta, delta) in milliseconds</th>
<th>37</th>
<th>43</th>
<th>51</th>
<th>63</th>
<th>82</th>
</tr>
</thead>
</table>
CONCEALABLE TIMING CHANNEL DECODER

Experiments:
- Purdue ➔ Princeton
- Telnet (i.i.d. Pareto)
- Data rate: 5 bits/sec
- Error rate: 1%

Receiver: \(R(1) R(2) R(3) R(4) \ldots R(2n-1) R(2n)\)

1: Inter-Reception Time Transformation:
\[ r^*(i) = \Gamma(R(i)) \]
\(\Gamma(x)\): CDF of a normal traffic

2: UnMasking Code.
   a) Same CSPRNG ➔ \(u(1), u(2), \ldots u(2n)\)
   b) \(x^*(i) = r^*(i) \oplus (1 - u(i))\), for \(i = 1, \ldots, 2n\)

   - \((x^*(1), x^*(2)) \longrightarrow c^*(1)\)
   - \((x^*(2i-1), x^*(2i)) \longrightarrow c^*(i)\)

Decoded Message: \(c^*(1) c^*(2) c^*(3) \ldots c^*(n)\)
SECURE ENCODER

- **Step 1: one-time pad**
  - Crypto Secure Pseudo Random Number Generator
    - Uniform (0,1): u(1), u(2), u(3),…
    - Symbol masking: r(i) = x(i) + u(i) mod 1
    - r(1), r(2), … are i.i.d. uniform random variables on (0,1)

- **Step 2: Getting desired statistical property**
  - T(i) = F^{-1}(r(i))

- **Claim:** T(1), T(2), … is computational indistinguishable from a normal traffic with distribution F(x)
SKETCH OF PROOF

- Proof by contradiction:
  - Assume Q, a polynomial time algorithm, can tell T(1), T(2), ... and a true sequence of i.i.d. random variable with c.d.f. F(x) apart
  - Can construct Q*, another polynomial time algorithm based on Q, to tell u(1), u(2), ... and a true i.i.d. uniform random variable apart.
  - Contradiction! Because u(1), u(2), ..., are crypto secure PRNG.
**Motivations**

- How fast can information be leaked through network covert timing channel?
  - on-off scheme: 17 bits/sec by Cubak, et al.
  - keyboard jitter bug: slow???

- Can we design a network timing channel that is impossible to detect?
## SUMMARY OF Decoding Error

Current Result (ccs’04):
- Data rate: 17 b/s
- error rate: 2%

<table>
<thead>
<tr>
<th>Δ (ms)</th>
<th>δ (ms)</th>
<th>data rate (bits/sec)</th>
<th>Princeton mean(%)</th>
<th>std dev (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>10</td>
<td>36.85</td>
<td>0.82</td>
<td>0.12</td>
</tr>
<tr>
<td>50</td>
<td>5</td>
<td>42.92</td>
<td>6.15</td>
<td>3.10</td>
</tr>
<tr>
<td>40</td>
<td>10</td>
<td>42.75</td>
<td>0.82</td>
<td>0.11</td>
</tr>
<tr>
<td>40</td>
<td>5</td>
<td>51.14</td>
<td>5.12</td>
<td>1.88</td>
</tr>
<tr>
<td>30</td>
<td>10</td>
<td>50.90</td>
<td>1.46</td>
<td>0.50</td>
</tr>
<tr>
<td>30</td>
<td>5</td>
<td>63.24</td>
<td>5.00</td>
<td>1.24</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>62.87</td>
<td>2.59</td>
<td>0.55</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>84.15</td>
<td>5.72</td>
<td>1.47</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>82.21</td>
<td>4.06</td>
<td>1.00</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>124.28</td>
<td>6.16</td>
<td>1.49</td>
</tr>
</tbody>
</table>

Average RTT (ms) 39.96
TIMING CHANNEL SOFTWARE

- Implementation:
  - Java Client/Server
  - Shared codebook (8-bits to 3-packets)
  - One way channel: no feedbacks from receiver
  - No need for time synchronization
  - Decoding errors do not propagate

- Deployment and Experiments:
  - Sender (Server) is deployed on a Purdue host
  - Receivers (Client) are deployed on PlaneLab nodes
OPTIMAL DATA RATE

Data Rate of Geometric Codes (bits/sec), $\Delta = 50$ ms

- $\delta = 10$ ms (solid blue line)
- $\delta = 5$ ms (dashed green line)
CONCEALABLE TIMING CHANNEL

Advantages:
- Immune from current and future detection
- Same codebook for different traffic patterns.
- No handshaking needed

Design Goals:
- Mimics statistical property
- Indistinguishable from normal traffic (computationally)