Leaking Information through Covert Timing Channel

Saurabh Bagchi

School of Electrical and Computer Engineering
Purdue University

Joint work with: Sarah Sellke, Ness Shroff, Chih-Chun Wang

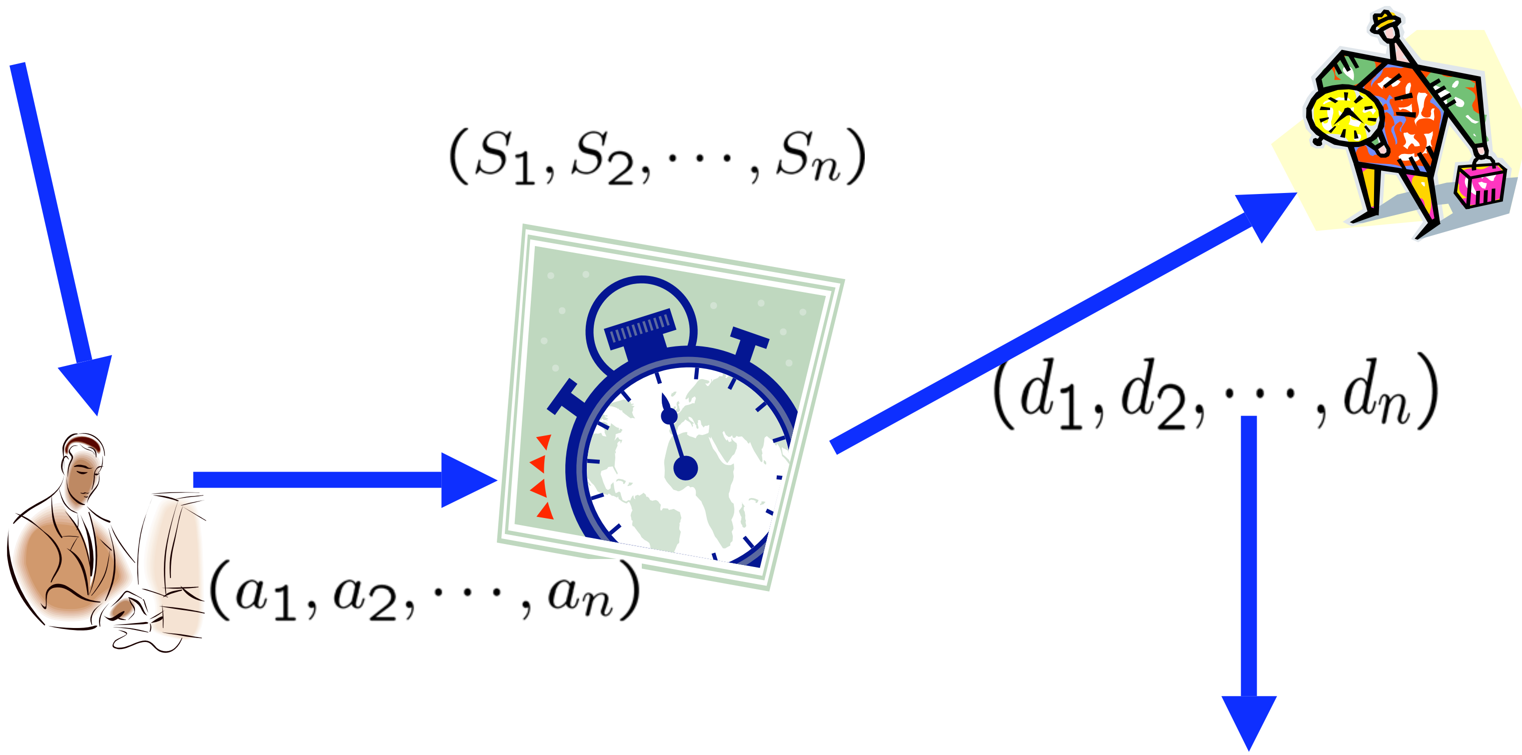
Work supported by:
NSF, Indiana 21st Century, Avaya, Motorola

A Brief History of Me

- 1996-2001: MS/PhD student in Computer Science, University of Illinois at Urbana-Champaign
  - Advisor: Ravi Iyer and Zbigniew Kalbarczyk
  - Thesis: Distributed Error Detection in Software Implemented Fault Tolerance Middleware (Chameleon)
- 2002-present: Assistant Professor in the School of Electrical and Computer Engineering, Purdue University
  - Courtesy Appointment in Computer Science
  - Group with 5 PhD students
- Attended & presented at FTCS/DSN in 1999, 2002-now
  - PDS PC member 2003-now

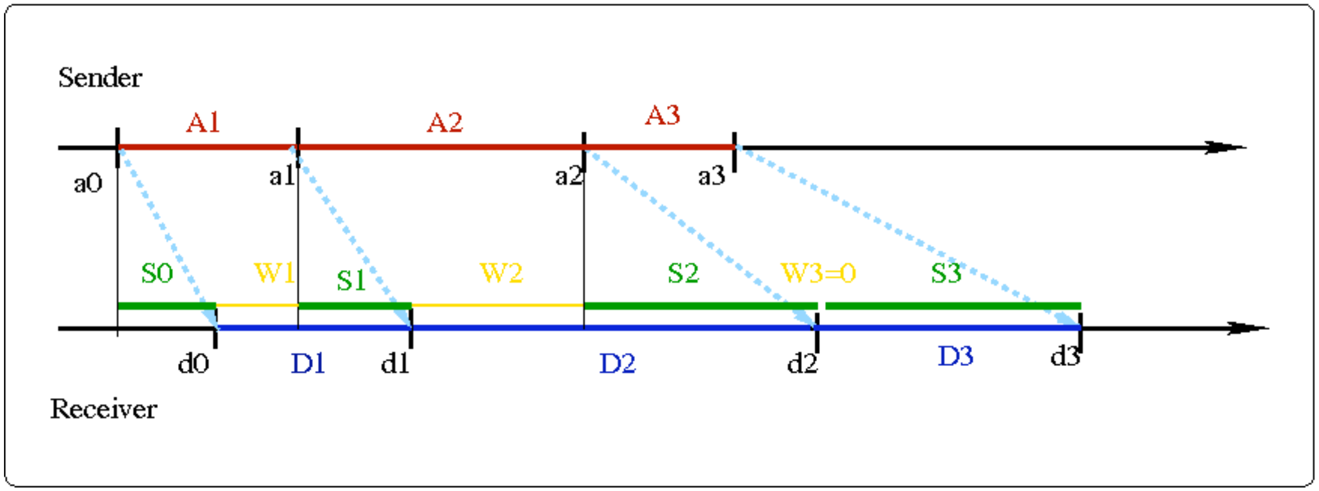
What are Timing Channels?

Msg(k)=00110100101



00110100101

Timing Channels



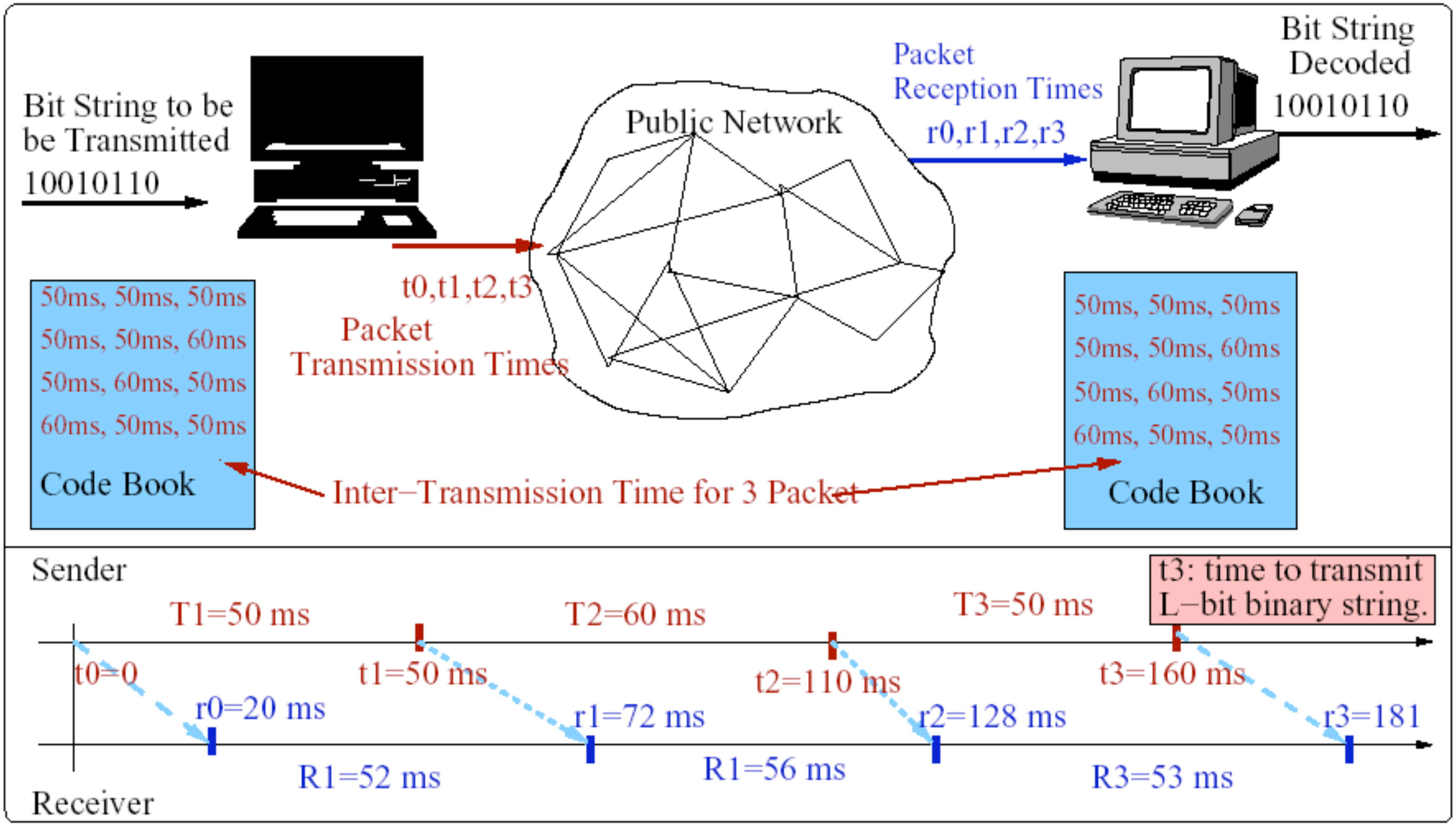
– Information is conveyed in the timing of the bits

- Sender: a_0, a_2, …, $a_{n-1.}$
- Server: S_0,S_2, …, S_{n-1}
- Receiver: d_0, d_1, …, d_{n-1}; and recovers information.

Network Timing Channels

- Implementing timing channels over a shared network between two distant computers is challenging
- Network timing channels are inherently noisy due to the delay and jitter in networks, which cause the timing information to be distorted when it reaches the receiver
- We use a (L-bit, n-packet) encoding: Encode L-bit binary strings in a sequence of n packet inter-transmission times $T_1, T_2, \cdots, T_n$
- Two objectives:
  - Increase data rate
  - Avoid detection of the channel

Illustration of Network Timing Channel



Problems Due to Variability in Channel

- Delay of a packet comprises fixed delay (*D*) and jitter (i.e., variability in delay) (ε)
- Say, D=30 ms, ε_{max}=5 ms
- Encoding:
  - "00" as t_0=0 ms, t_1=60 ms (T_1=60 ms)
  - "11" as t_0=0 ms, t_1=68 ms (T_1=68 ms)
- Reception:
  - "00" as r_0=31 ms (ε=1 ms), r_1=94 ms (ε=4 ms)
  - "11" as r_0=31 ms (ε=1 ms), r_1=94 ms (ε=-4 ms)
  - Cannot distinguish between the two

Design Parameters of the Covert Channel

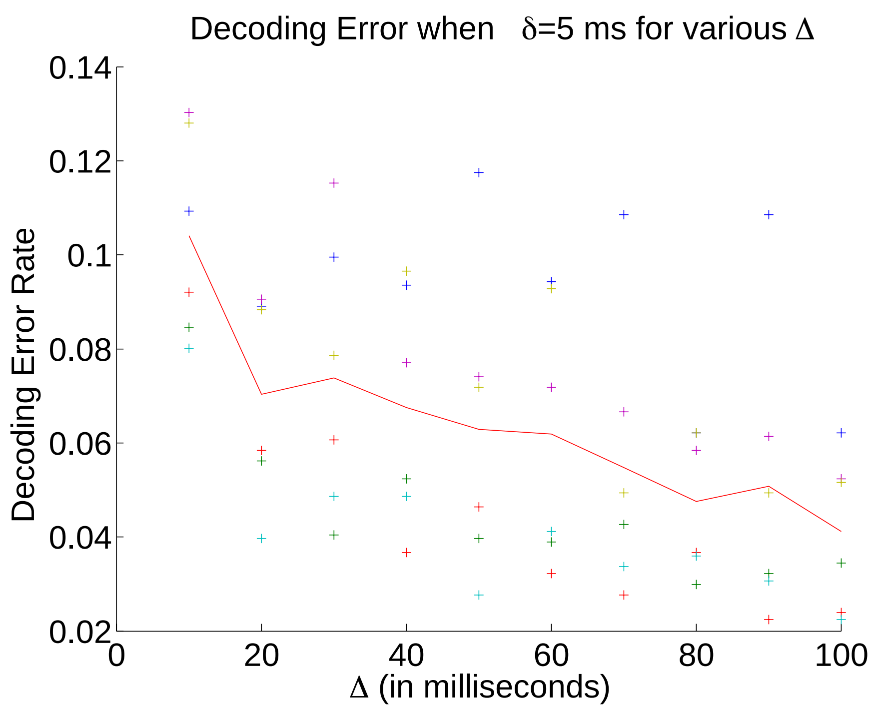
- Minimum difference between two inter-transmission times representing two different code words: δ
- Result: For proper decoding $\delta > 4\varepsilon_{max}$
- The minimum value for inter-transmission time: Δ
- If packets transmitted too close to each other, then queuing may result destroying timing information
- L-bits to n-packets encoding: Geometric codes used where $T_i = \Delta + k_i\delta$
  - $K = \sum_{i=1}^{n} k_i$
- Rate of channel is non-monotonic wrt K

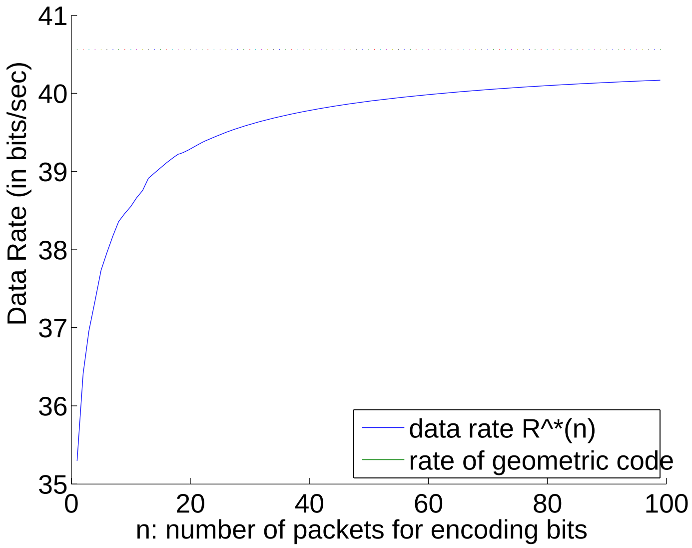
Covert Timing Channels in Practice

- Practical Design and Implementation of a covert timing channel over TCP/IP networks.
- Improvement over state-of-the-art:
  - No necessity for synchronization, feedback
  - No error propagation
- Code: 8-bit ascii code mapped to 3 packets
  - Δ=50 ms, δ=10 ms
  - Example: '!' mapped to (T1, T2, T3) = (50, 80, 100) ms
- Experiments on computers at Purdue and Princeton
  - Network Delay Characteristics: RTT = 40 ms, small Jitter (3-5%)
- Rate of the TCP/IP Timing Channel:
  - Up to 80 bit/sec, 5 times improvement over the on-off channels with comparable error rate
- By introducing random delay in inter-transmission time using a key agreed upon by sender and receiver, the covert traffic can match any normal traffic pattern

Results

Δ (ms)	Factor improvement in data rate
20	5.05
40	3.07
60	2.21
80	1.73
100	1.42





- Tradeoff between decoding error and channel rate
  - Can be mitigated by error correcting code
- We can come close to the theoretical maximum achievable on the channel

What next?

- We showed a way of practically implementing covert timing channel
  - Higher data rate achieved
  - Can mimic normal traffic
- ToDo
  - Detection mechanism
  - Using covert channel for key distribution
- Publications:
  - Allerton Communication Conference, 2006
  - Information Theory Conference, 2007