TCP/IP Covert Timing Channel: Theory to Implementation

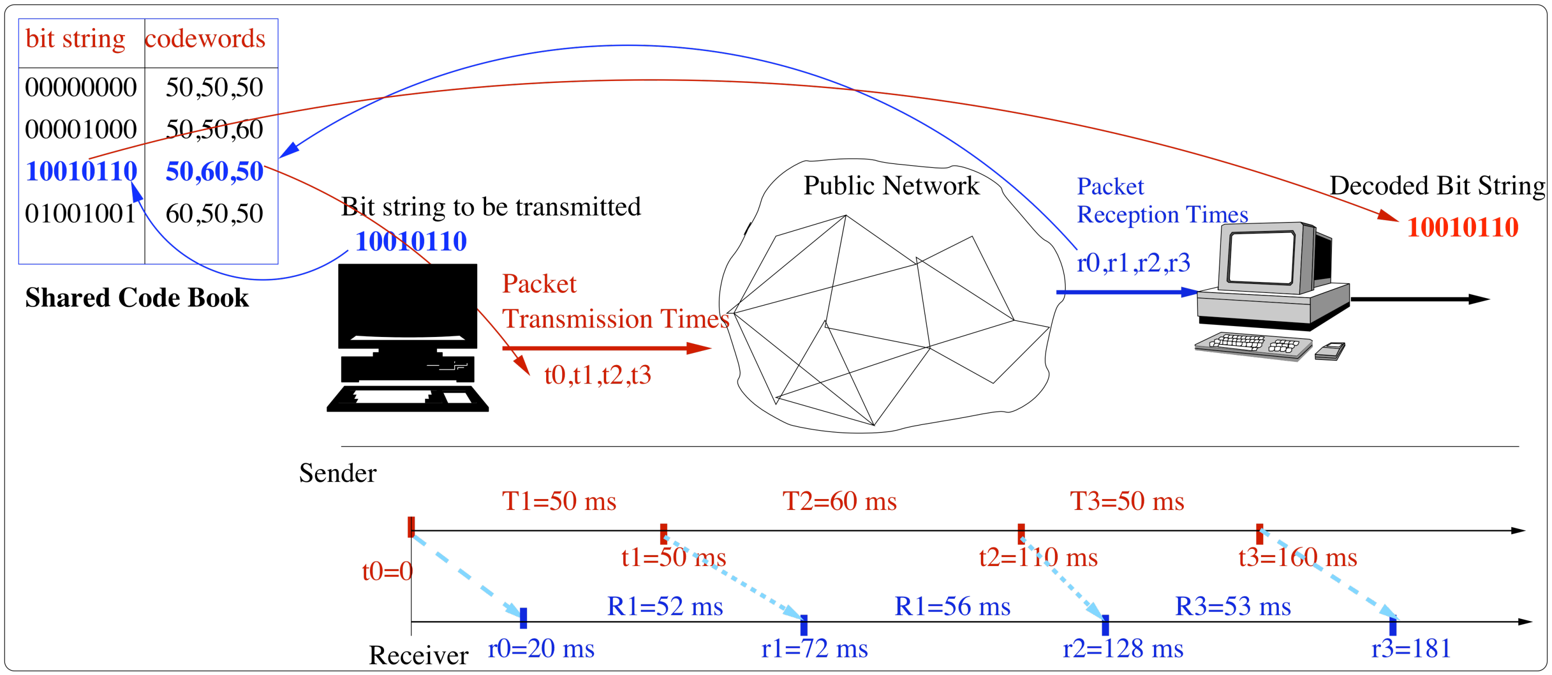
Sarah H. Sellke, Chih-Chun Wang
Saurabh Bagchi, and Ness B. Shroff

Purdue University

The Ohio State University

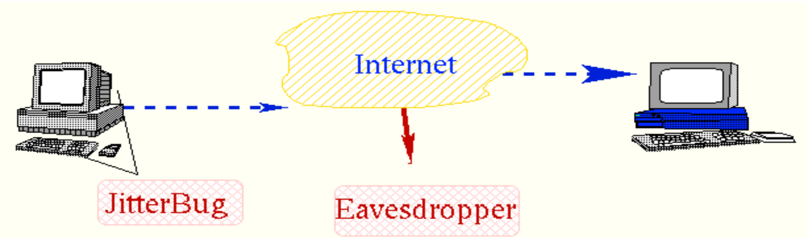
Network Covert Timing Channels

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 <u>two</u> 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 : Δ and δ
  - A 4-bits to 2-packets scheme (Δ=60 ms, δ=10 ms)
  - T1, T2: packet inter transmission times

Bit String	0000	0001	0010	0011	0100				1111
(T1, T2)	(60,60)	(60,70)	(70,60)	(70,70)	(60,80)				(100,100)

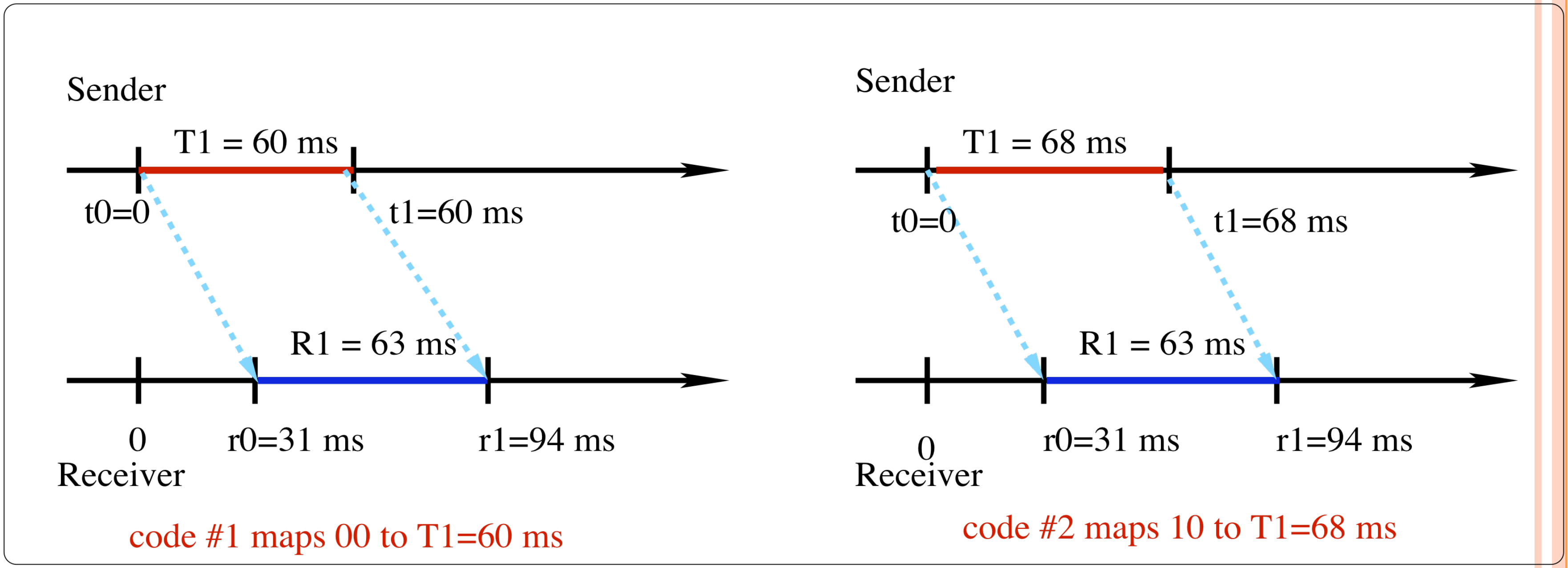
  - T1, T2, T3, …, Tn takes values from the set

$$E = \{T: T=\Delta+k*\delta, k=0, 1, 2, \ldots\}$$

The Ohio State University
Purdue University

Example of decoding error

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



Design Challenge

- Determine the optimal values of L and n
- Two simple examples (Δ=60 ms, δ=20 ms):
  - 2-bits to 1-packets scheme: 22 bits/sec

Bit strings	**00**	**10**	**01**	**11**
T1	60	80	100	120

  - 4-bits to 1-packets scheme: 19 bits/sec

Bit strings	**0000**	**1001**	**…**	**1111**
T1	60	80	…	360

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
  - $T(i) = \Delta + k(i) * \delta$
  - K: $k(1)+k(2)+\ldots+k(n) \leq K$
  - 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 (Δ, δ), the data rate R(n,K) of an (n, K)-code

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

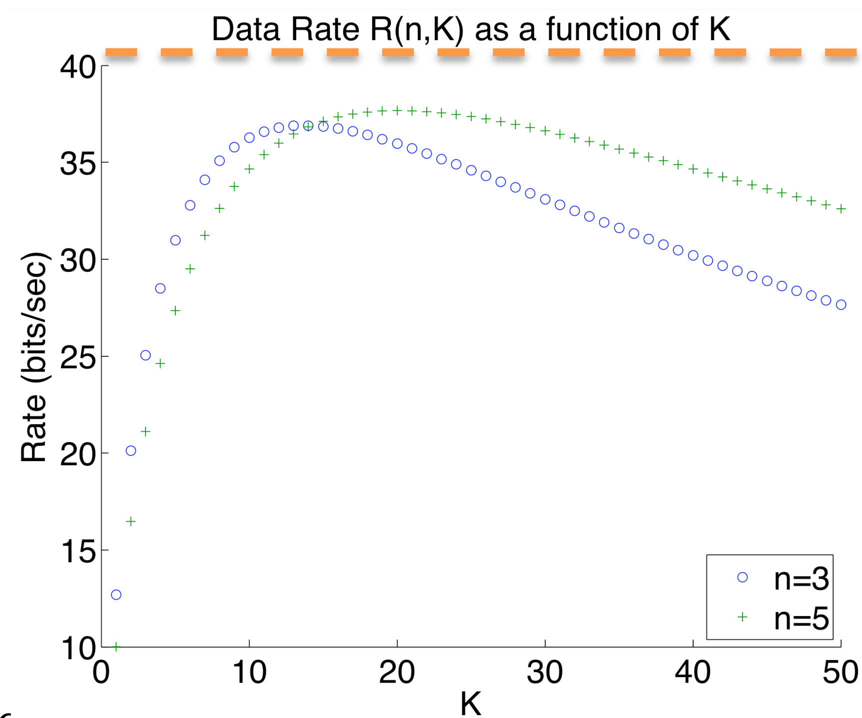
- Main Result:
  - Optimal Data Rate R*(n) given (Δ, δ):

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

THE OHIO STATE UNIVERSITY PURDUE UNIVERSITY

PLOT OF DATA RATE R(n,K)

- Δ=50 ms, δ=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

THE OHIO STATE UNIVERSITY PURDUE UNIVERSITY

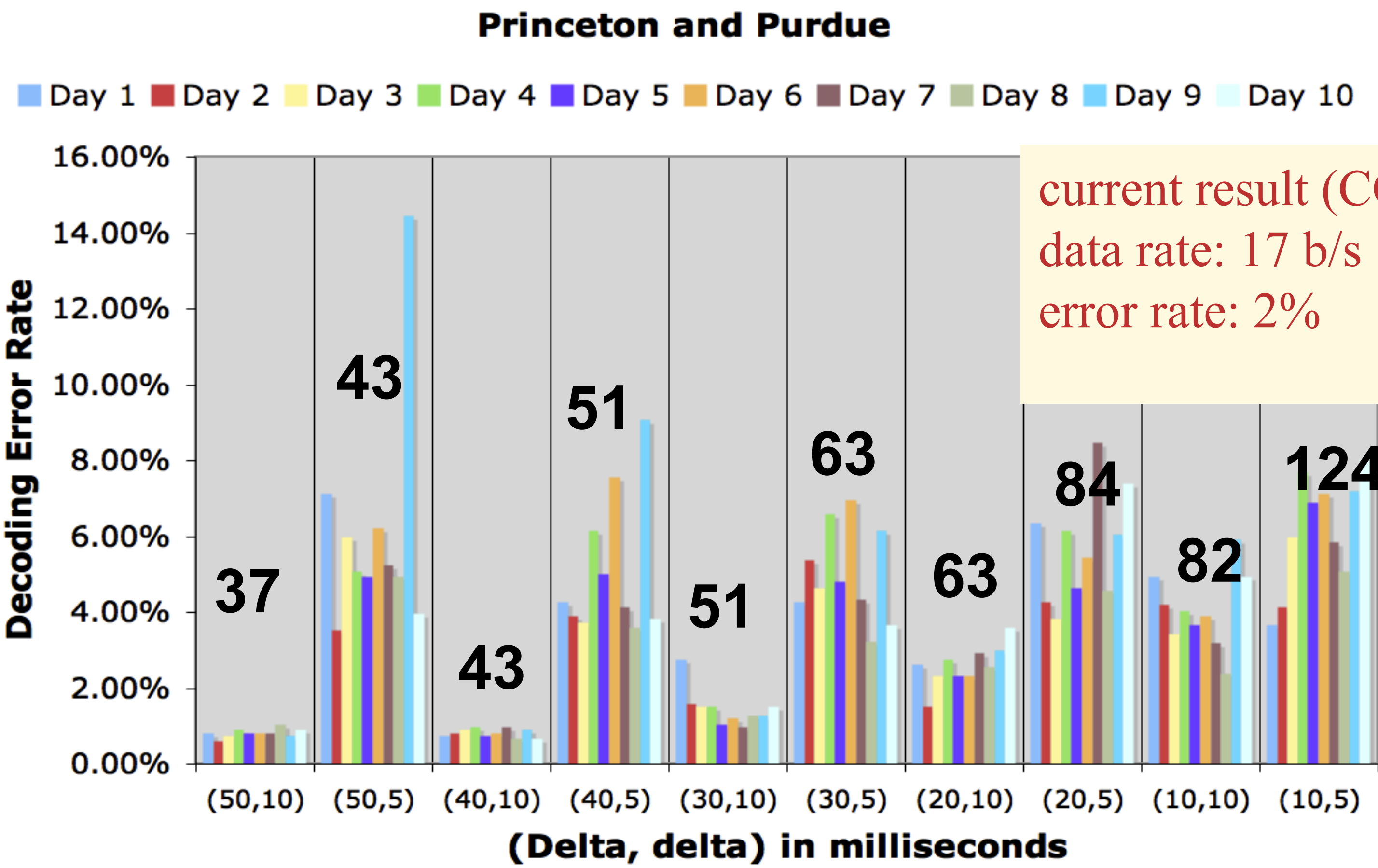
OUTLINE

- Design of Timing Channel 1
- Experimental Results
- Concealable Timing Channels

THE OHIO STATE UNIVERSITY PURDUE UNIVERSITY

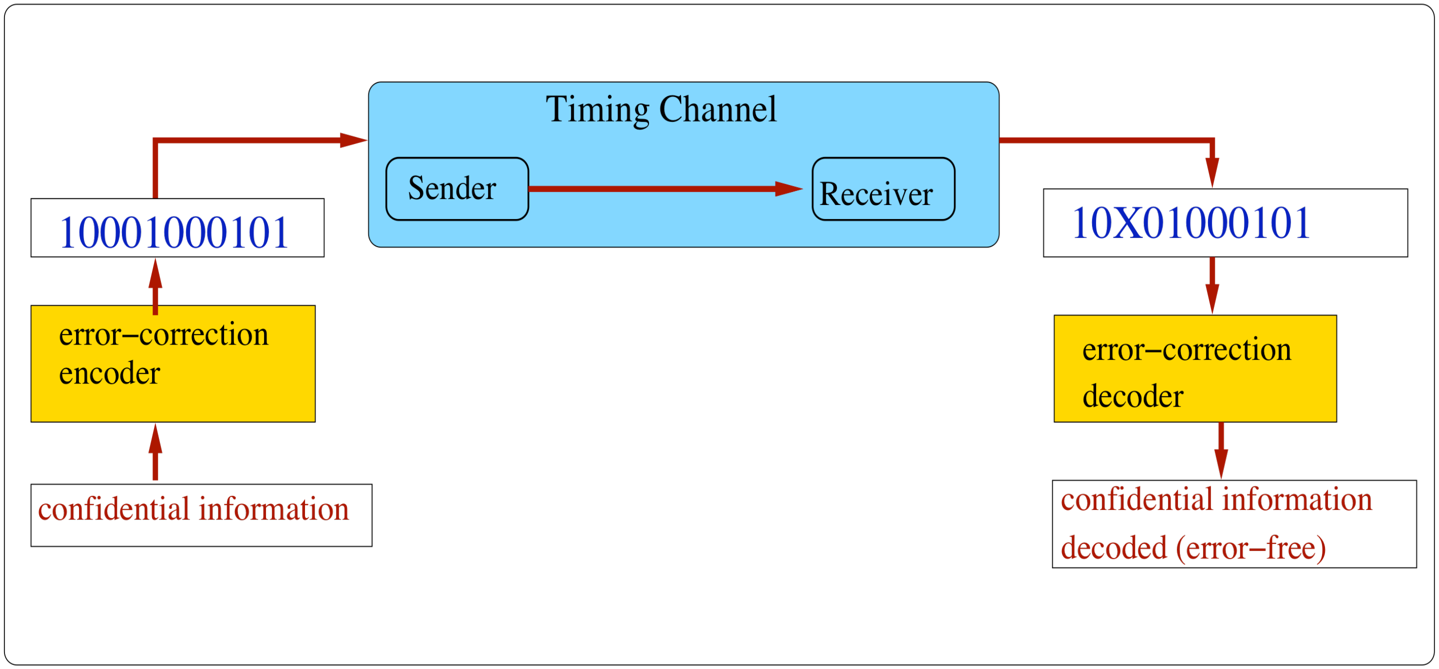
EXPERIMENTS

DECODING ERRORS



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

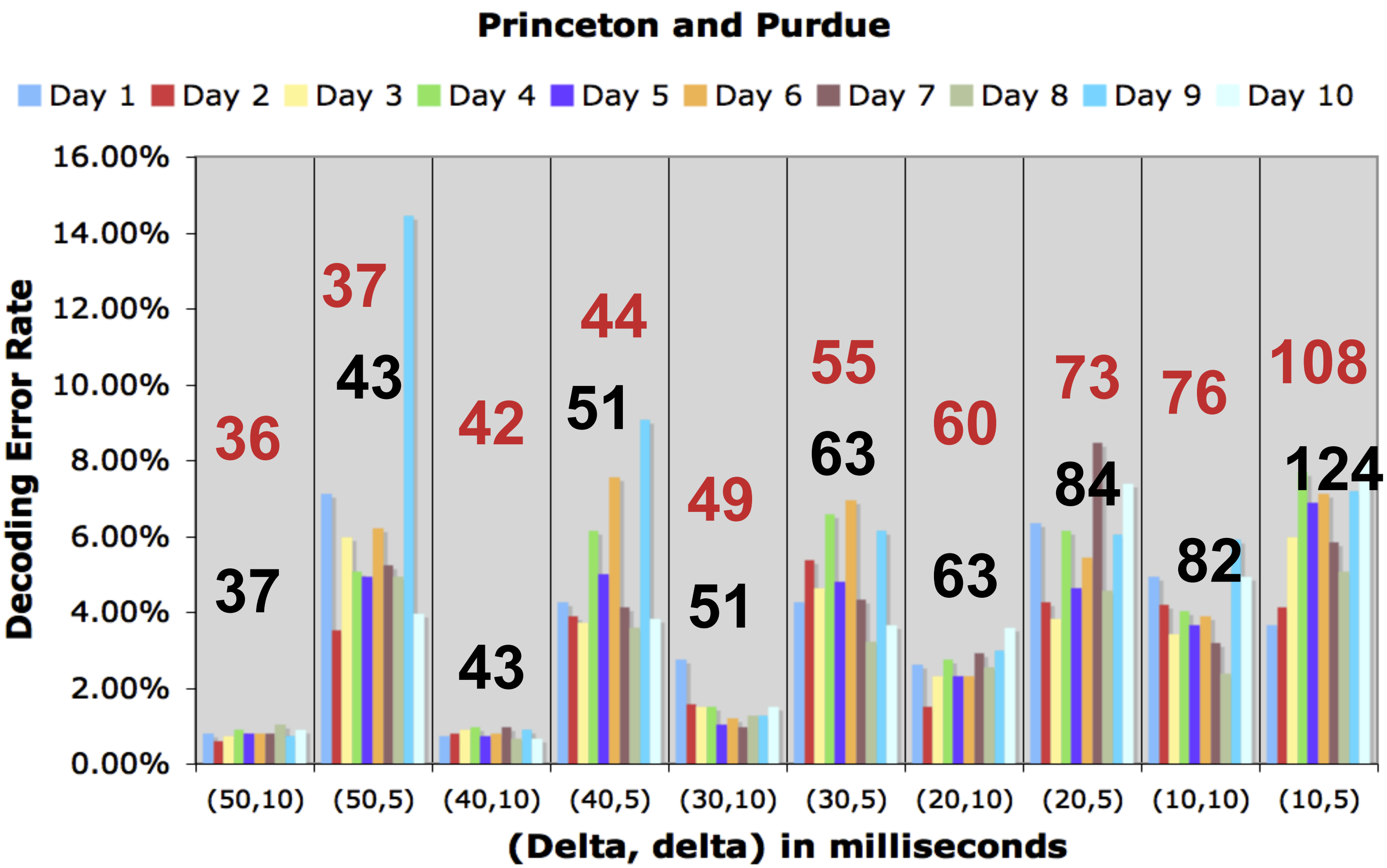
ERROR CORRECTION



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

THE OHIO STATE UNIVERSITY PURDUE UNIVERSITY

DECODING ERRORS



OUTLINE

- Design of Timing Channel 1
- Experimental Results
- Concealable Timing Channel

TYPE 2 TIMING CHANNEL: CONCEALABLE

- Goal:
  - Immune against current and future detection

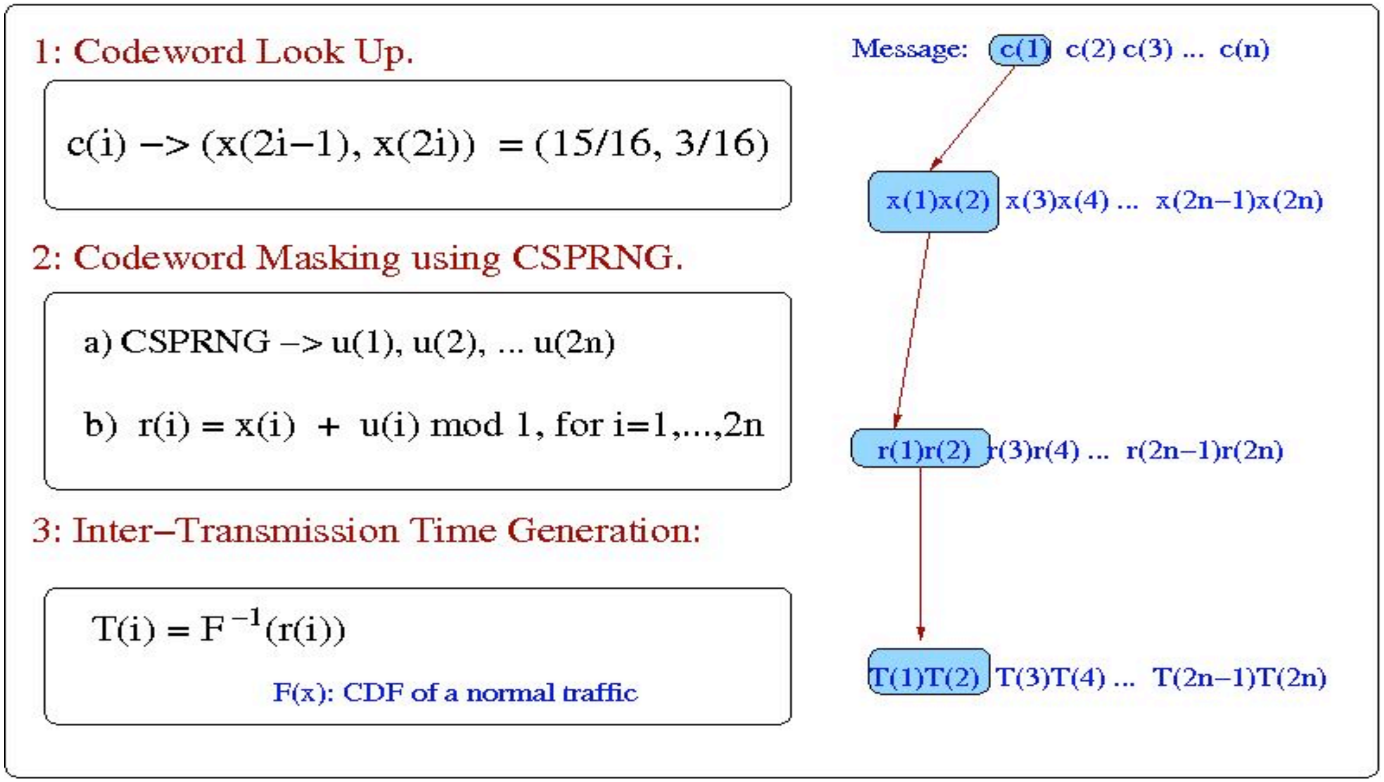
- How do we achieved this goal?
  - Mimic the statistical property of i.i.d. normal traffic
  - Computationally indistinguishable from i.i.d. normal traffic

- Timing channel is a serious security concern

Concealable Timing Channel

Achieving Design Goals:

- Mimics statistical property
- Computationally indistinguishable from i.i.d. normal traffic



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%

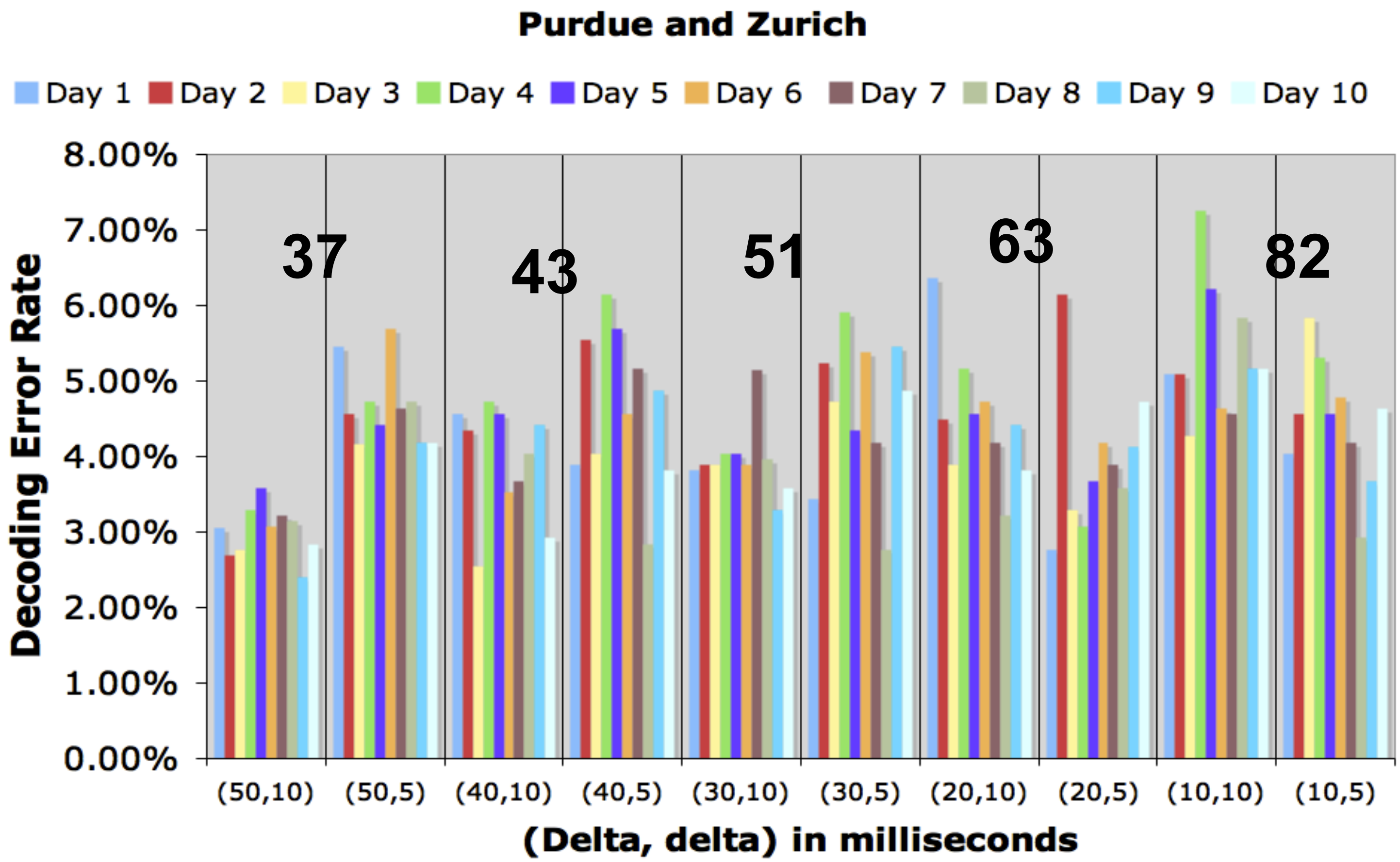
CONCLUSION

- 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)

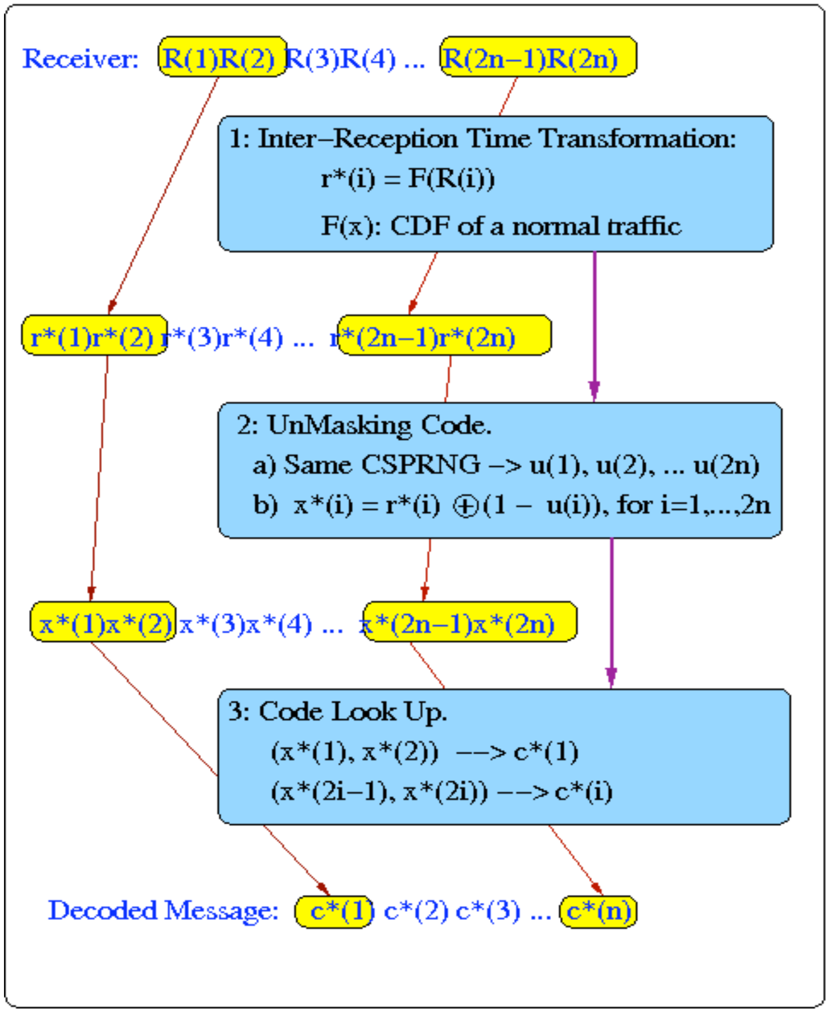
- Future Direction:
  - Efficient algorithm to mimic correlated traffic, such as HTTP traffic

Thank You!

DECODING ERRORS



Concealable Timing Channel Decoder



Experiments:

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

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)

THE OHIO STATE UNIVERSITY PURDUE UNIVERSITY

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

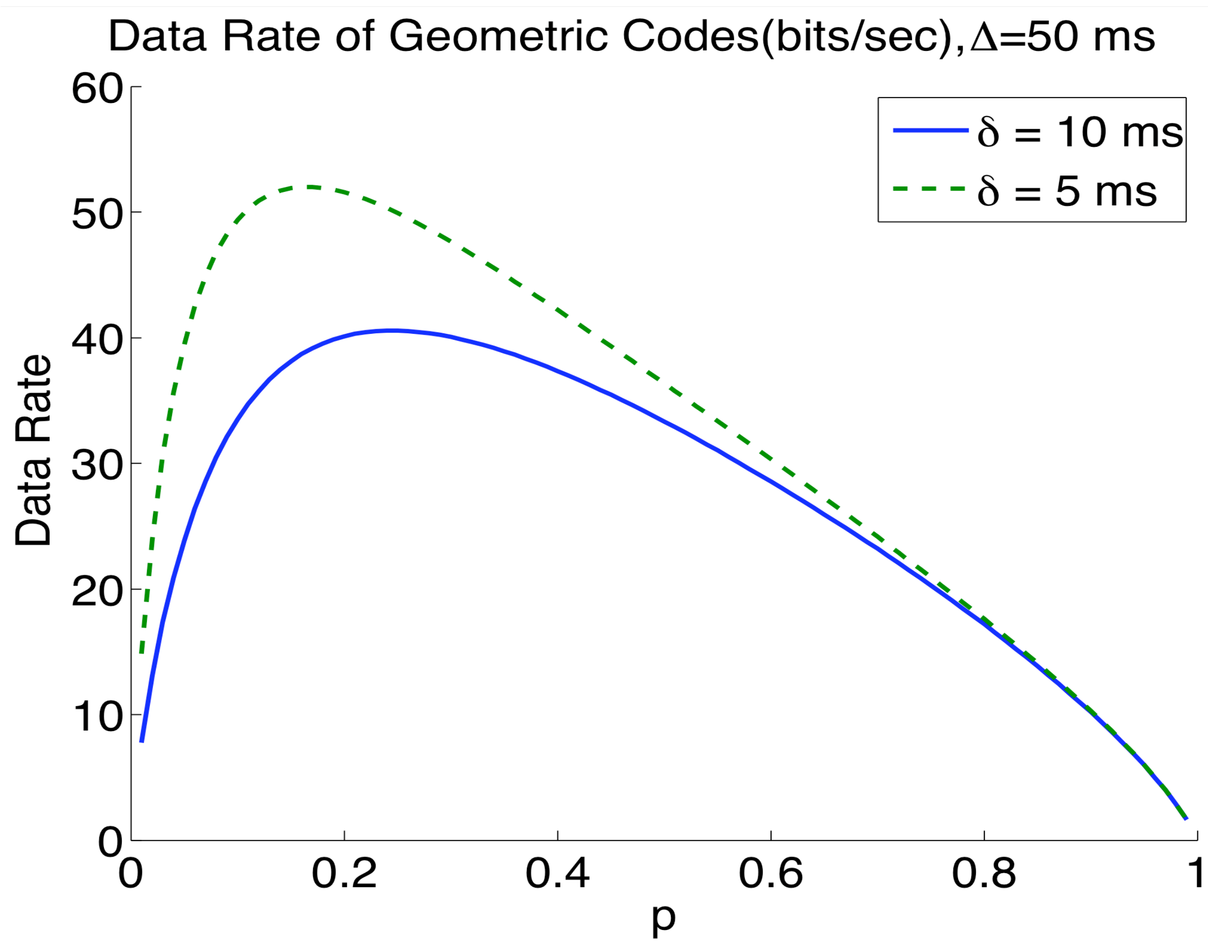
Δ (ms)	δ ms	data rate (bits/sec)	Princeton mean(%)	stdev (%)
50	10	36.85	0.82	0.12
50	5	42.92	6.15	3.10
40	10	42.75	0.82	0.11
40	5	51.14	5.12	1.88
30	10	50.90	1.46	0.50
30	5	63.24	5.00	1.24
20	10	62.87	2.59	0.55
20	5	84.15	5.72	1.47
10	10	82.21	4.06	1.00
10	5	124.28	6.16	1.49
		Average RTT (ms)		39.96

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

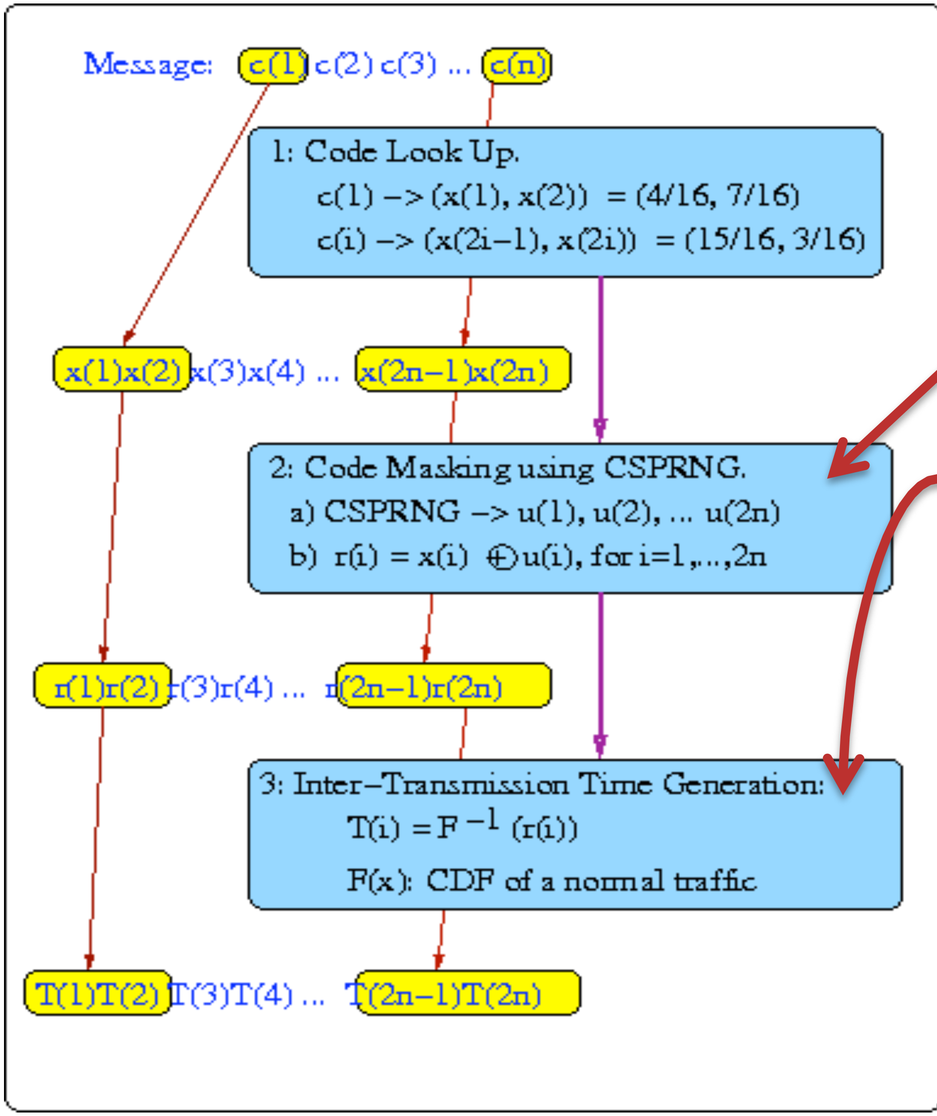
TIMING CHANNEL SOFTWARE

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

OPTIMAL DATA RATE



Concealable Timing Channel



Design Goals:

- **Mimics statistical property**
- **Indistinguishable from normal traffic (computationally)**

Advantages:

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