# Statistical Measurement of Information Leakage

Tom Chothia
Univ. of Birmingham

Joint work with Kostas Chatzikokolakis and Apratim Guha

# Introduction

- Some background on anonymity.
- Information theory for security and anonymity.
- Estimate information leakage from trial runs of a real system.
  - Automatic tool to calculate information leakage.
  - We present an ***if, and only if***, test for ***zero*** leakage.

# Anonymity Set

- Measure of anonymity is the size of the set of possible IDs
- If you know its one of 10 people set size = 10
- If are 99.9999% sure it's one person, but it may also be 1 of 99 others then set size = 100 !!

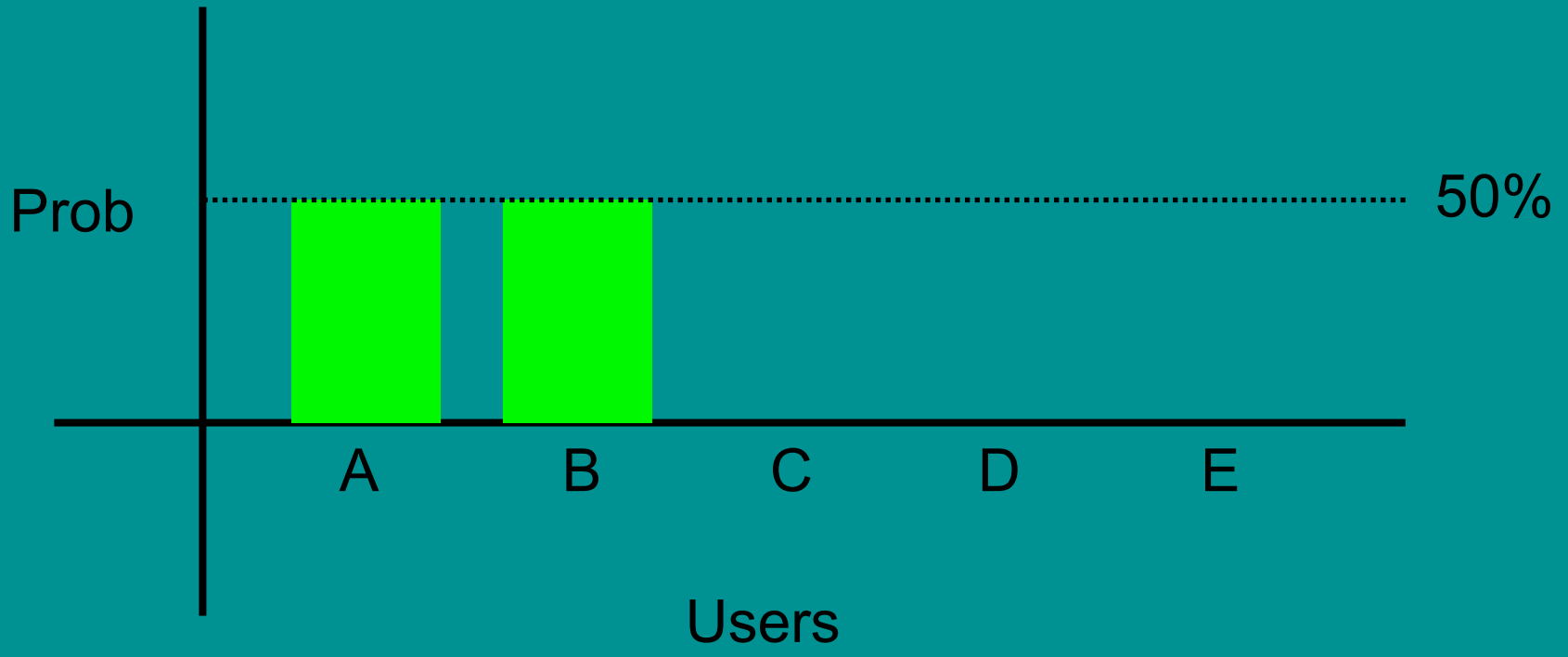
# Levels of Anonymity

Reiter and Rubin provide the classification:

- ***Beyond suspicion***: the user appears no more likely to have acted than any other.
- ***Probable innocence***: the user appears no more likely to have acted than to not to have.
- ***Possible innocence***: there is a nontrivial probability that it was not the user.

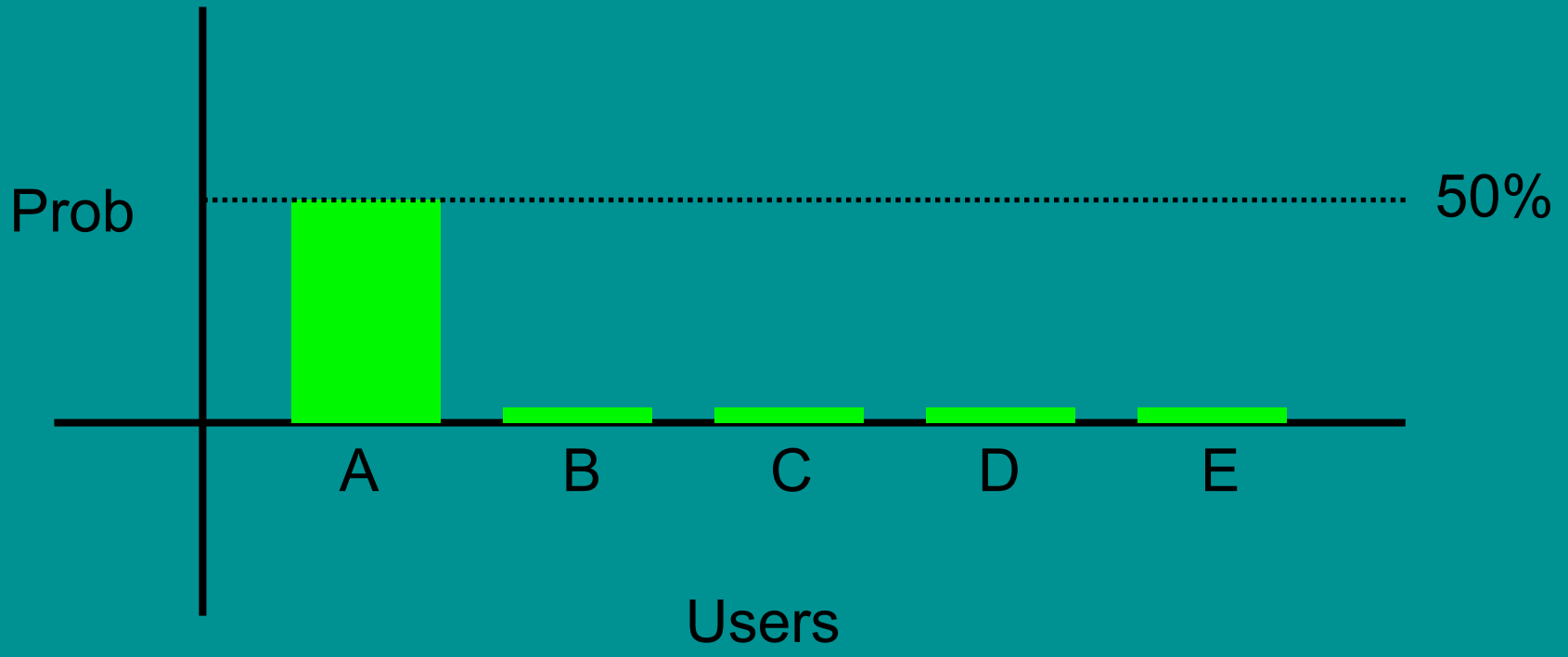
# Probable Innocence

- After a run of the system, who does the observer think did it?



# Probable Innocence

- After a run of the system, who does the observer think did it?



# Information Theory

$$H(X) = -\sum_{x \text{ in } X} p(x).\log(p(x))$$

Entropy describes the “amount of chaos” or uncertainty in a system

= the number of bits needed to describe the system, on average.

# The Horse Race Example

- Race 1: 8 horses, all equally likely to win the race ... very chaotic.

How many bit on average do you need to send the identify of the winner?

# The Horse Race Example

- Race 1: 8 horses, all equally likely to win the race ... very chaotic.

$$H(X) = -( 0.125.\log(0.125) + ... + )0.125.\log(0.125) )$$
$$= \log(0.125) = 3$$

i.e. on average you need three bits of information to send the identify of the winner.

# The Horse Race Example

Race 2: 8 horse:

- P(horse1 wins) = 1/2
- P(horse2 wins) = 1/4
- P(horse3 wins) = 1/8
- P(horse4 wins) = 1/16
- P(horse5 wins) = 1/64
- P(horse6 wins) = 1/64
- P(horse7 wins) = 1/64
- P(horse8 wins) = 1/64

The result is much more predicable, much less chaos.

# The Horse Race Example

Race 2: 8 horse:

- P(horse1 wins) = 1/2
- P(horse2 wins) = 1/4
- P(horse3 wins) = 1/8
- P(horse4 wins) = 1/16
- P(horse5 wins) = 1/64
- P(horse6 wins) = 1/64
- P(horse7 wins) = 1/64
- P(horse8 wins) = 1/64

The result is much more predicable, much less chaos.

H(X) = 1/2.log(1/2)
+ 1/4.log(1/4) +
....+1/64/.log(1/64)
= 2

i.e. on average we need to send 2 bits

# The Horse Race Example

Race 2: 8 horse:

- P(horse1 wins) = 1/2
- P(horse2 wins) = 1/4
- P(horse3 wins) = 1/8
- P(horse4 wins) = 1/16
- P(horse5 wins) = 1/64
- P(horse6 wins) = 1/64
- P(horse7 wins) = 1/64
- P(horse8 wins) = 1/64

The result is much more predicable, much less chaos.

H(X) = 1/2.log(1/2)
+ 1/4.log(1/4) +
....+1/64/.log(1/64)
= 2

i.e. on average we need to send 2 bits

1->0, 2->10, 3->110,
4->1110, 5->111100,
6->111101, 7->111110
8->111111

# Information Theory

$$H(X) = -\sum_{x \text{ in } X} p(x).\log(p(x))$$

Entropy describes the “amount of chaos” or uncertainty in a system

= the number of bits needed to describe the system, on average.

# A Metric For Anonymity

- The entropy of the set of possible people.
- For anonymity proposed independently in 2002 by Daiz et al. and Danezis et al.
- But what about
  - user actions?
  - attack has some prior knowledge?
  - some users are more likely to be guilty than others?

# Conditional Entropy

Conditional Entropy H(Y|X) is the remaining chaos in Y once you know X:

$$H(Y|X) = \sum_x p(x).\ H(Y|X=x)$$
$$= -\sum_{x,y} p(y,x).\log(\ p(y|x)\ )$$

- if you're sending X then H(Y|X) is the average no. of bits needed to send Y as well.
- Proposed for security by McIver and Morgan in 2003

# Mutual Information

Mutual Information I(X;Y) is the reduction of uncertainty you get in X if you know Y.

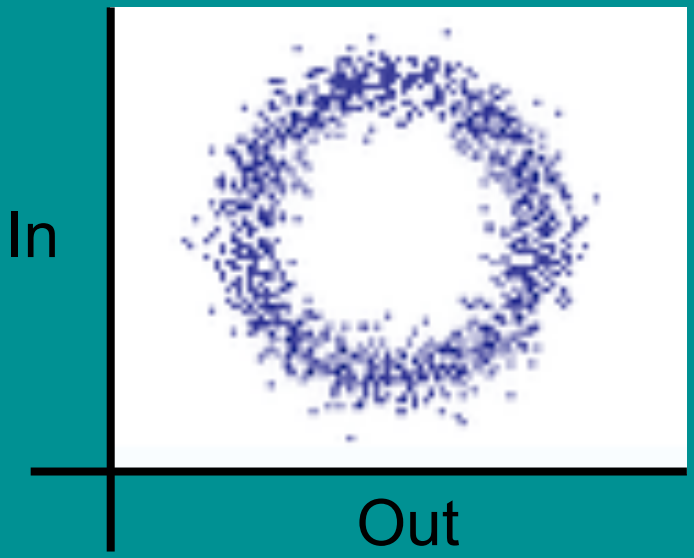
$$I(X;Y) = H(X) - H(X|Y)$$
$$= H(Y) - H(Y|X)$$

If W gives the conditions probabilities of Y given X we also write:

$$I(Q,W) = I(Q;QW)$$
$$= \sum_{x,y} Q(x).W(y|x).\log( W(y|x) / QW(y) )$$

# Mutual Information vs. Correlation

- Mutual Information measures any link in the data sets.
- Correlations only measures a linear link.




Corr = 0  M.I. ≠0

# Conditional Mutual Information

- Mutual Information can be conditional:

$$I(L_1;H \mid L_2) = H(L_1 \mid L_2) - H(L_1 \mid H,L_2)$$

This is the information you learn about H from L1, given you know L2.

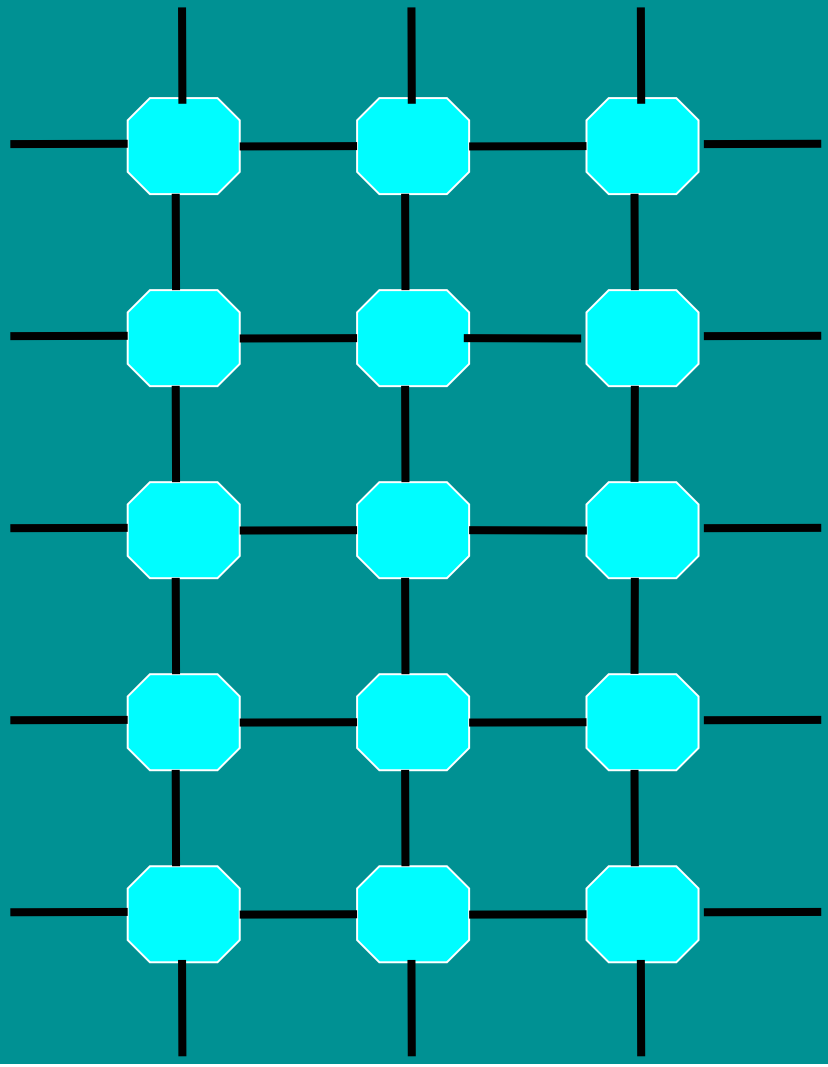
Used by Clark et al. for security.

# Assuming a Uniform Distribution Doesn't Work

Imagine a network of peers.

Each peer randomly picks a nieghbour to act as a proxy for each message.

Communication between peers is undetectable.

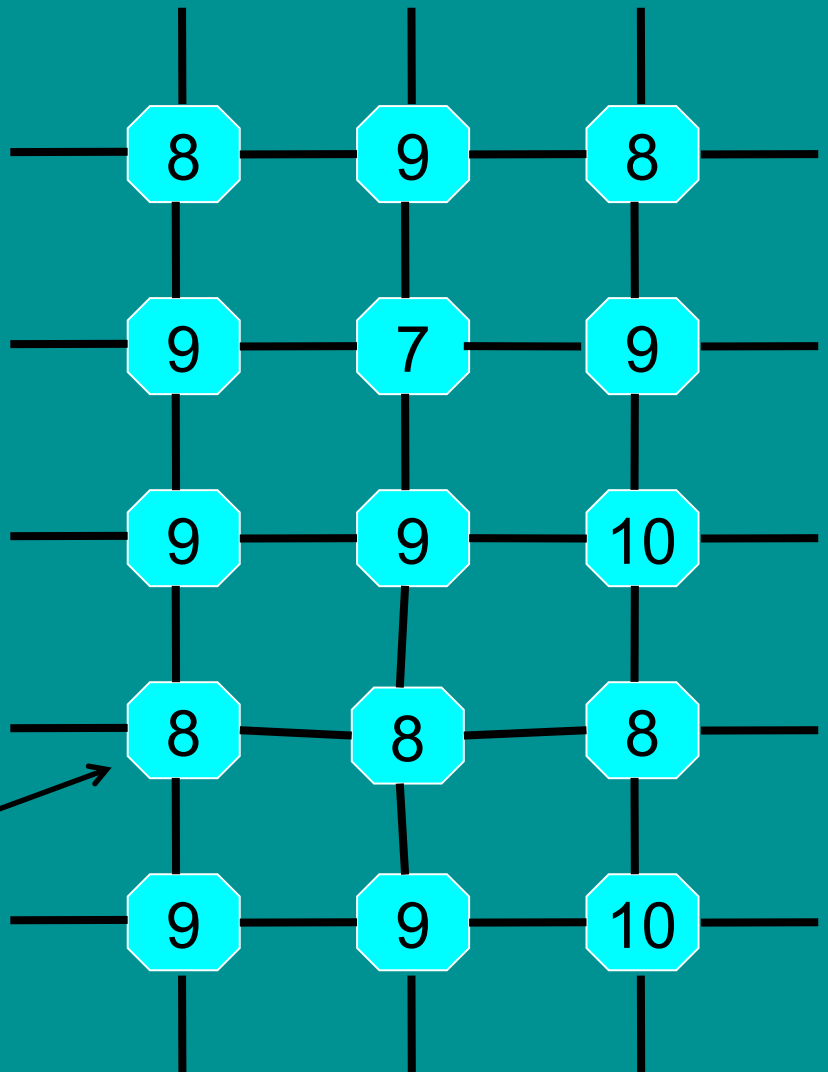


# Assuming a Uniform Distribution Doesn't Work

We can observe the messages leaving each peer.

If each peer sends uniformly then we have a 1 in 4 change of guessing the sender

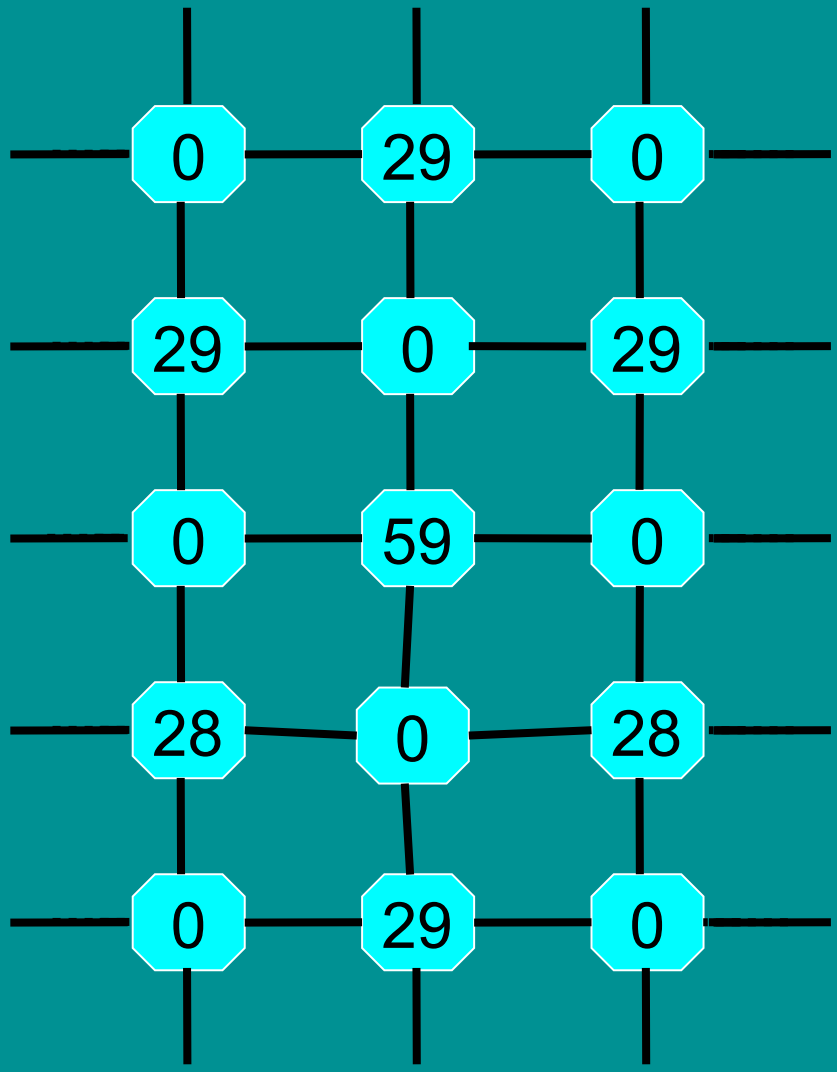
Number of observed messages from node



# Assuming a Uniform Distribution Doesn't Work

But if the distribution isn't uniform then a sender has nowhere near this anonymity.

In the worst case the anonymity is zero



# Channel Capacity

For a channel:

I(Inputs;Output) = how much the outputs tell you about inputs

The most information it is possible to send over a channel

$$C = \mathrm{Max}_{\mathrm{Inputs}} \; I(\mathrm{Inputs};\mathrm{Outputs})$$

# Information Leakage = Capacity (System)

Following Chatzikokolakis et al., Millen, Clark et al. etc.

- Think of the whole system as a channel.
  - secret is the input to the “channel”.
  - observables are the outputs from the “channel”.
- Capacity tells us what an attacker can learn about the users from the observable actions.

# Information Theory

**Entropy**: $H(X) = - \sum_x p(x).\log(p(x))$

the amount of uncertainty in X.

**Conditional Entropy**: $H(Y|X) = \sum_x p(x).H(Y|X=x)$

the amount of uncertain in Y if you know X

**Mutual Information**: $I(X;Y) = H(X) - H(X|Y)$

the reduce of uncertainty you get in X if you know Y.

**Relative Entropy**: $D(p||q) = \sum_x p(x).\log(p(x) / q(x))$

"distance" from one distribution to another.

$I(p(x),p(y)) = D(p(x,y) \,||\, p(x).p(y))$

# Information Theory

**Entropy**: $H(X) = - \sum_x p(x).\log(p(x))$

the amount of uncertainty in X.

**Conditional Entropy**: $H(Y|X) = \sum_x p(x).H(Y|X=x)$

the amount of uncertain in Y if you know X

**Mutual Information**: $I(X;Y) = H(X) - H(X|Y)$

the reduce of uncertainty you get in X if you know Y.

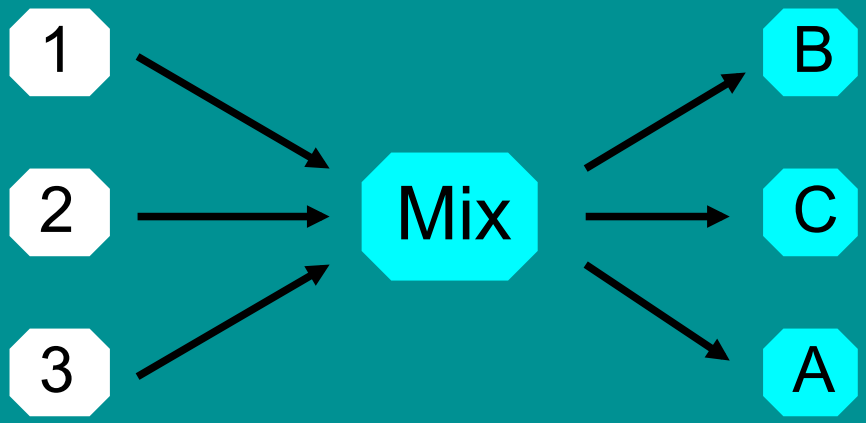
**Relative Entropy**: $D(p||q) = \sum_x p(x).\log(p(x) / q(x))$

"distance" from one distribution to another.

$I(p(x),p(y)) = D(p(x,y) \,||\, p(x).p(y))$

# MIXes

- MIXes are proxies that forward messages between them
- The MIX waits until it has received a number of messages, then forwards them in different order.
- E.g. 1 wants to send to A, 2  to B and 3 to C



# A Perfect Mix

| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 1,3,2 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 2,1,3 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 2,3,1 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 3,1,2 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 3,2,1 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |

(a) Probabilites of outputs for each input for a perfect mix node

# A Perfect Mix

| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 1,3,2 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 2,1,3 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 2,3,1 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 3,1,2 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |
| in 3,2,1 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 | 0.1666 |

(a) Probabilites of outputs for each input for a perfect mix node

Information Leakage = Capacity = 0

# A Bad Mix

| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0 | 0.3333 | 0.3333 | 0 | 0 | 0.3333 |
| in 1,3,2 | 0.3333 | 0 | 0 | 0.3333 | 0.3333 | 0 |
| in 2,1,3 | 0.3333 | 0 | 0 | 0.3333 | 0.3333 | 0 |
| in 2,3,1 | 0 | 0.3333 | 0.3333 | 0 | 0 | 0.3333 |
| in 3,1,2 | 0 | 0.3333 | 0.3333 | 0 | 0 | 0.3333 |
| in 3,2,1 | 0.3333 | 0 | 0 | 0.3333 | 0.3333 | 0 |

(b) Probabilites of outputs for each input for a flawed mix node

Information Leakage = Capacity = ?

# A Bad Mix

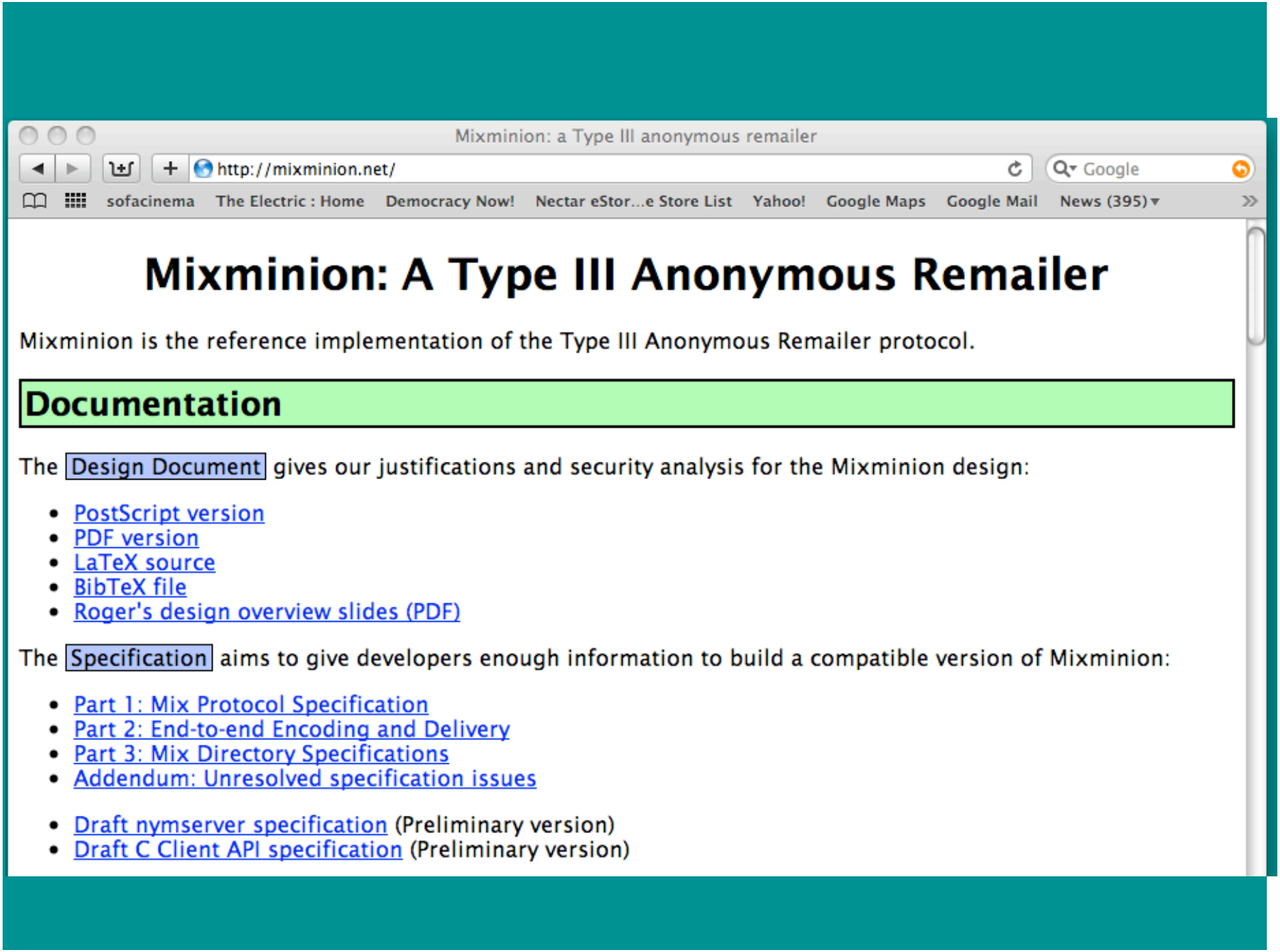
| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0 | 0.3333 | 0.3333 | 0 | 0 | 0.3333 |
| in 1,3,2 | 0.3333 | 0 | 0 | 0.3333 | 0.3333 | 0 |
| in 2,1,3 | 0.3333 | 0 | 0 | 0.3333 | 0.3333 | 0 |
| in 2,3,1 | 0 | 0.3333 | 0.3333 | 0 | 0 | 0.3333 |
| in 3,1,2 | 0 | 0.3333 | 0.3333 | 0 | 0 | 0.3333 |
| in 3,2,1 | 0.3333 | 0 | 0 | 0.3333 | 0.3333 | 0 |

(b) Probabilites of outputs for each input for a flawed mix node

Information Leakage = Capacity = 1 bit

# Applying this to Real Systems

- How do we apply information theoretic measures to real systems?
- Leak may be caused by the implementation:
  - Time based attack on RSA (Paul Kocher)
  - Bandwidth attack on Tor (Murdoch & Danezis)
  - CPU Heat attack on Tor Hidden services (Murdoch)


Mixminion: a Type III anonymous remailer
http://mixminion.net/
Google
sofacinema
The Electric : Home
Democracy Now!
Nectar eStor...e Store List
Yahoo!
Google Maps
Google Mail
News (395)
Mixminion: A Type III Anonymous Remailer
Mixminion is the reference implementation of the Type III Anonymous Remailer protocol.
Documentation
The Design Document gives our justifications and security analysis for the Mixminion design:
PostScript version
PDF version
LaTeX source
BibTeX file
Roger's design overview slides (PDF)
The Specification aims to give developers enough information to build a compatible version of Mixminion:
Part 1: Mix Protocol Specification
Part 2: End-to-end Encoding and Delivery
Part 3: Mix Directory Specifications
Addendum: Unresolved specification issues
Draft nymserver specification (Preliminary version)
Draft C Client API specification (Preliminary version)

# Prob. Observed from Mixminion Node

| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0.0 | 0.0118 | 0.0473 | 0.0118 | 0.0059 | 0.9231 |
| in 1,3,2 | 0.0117 | 0.0 | 0.0351 | 0.0292 | 0.0 | 0.924 |
| in 2,1,3 | 0.005 | 0.0222 | 0.0278 | 0.0444 | 0.0056 | 0.8944 |
| in 2,3,1 | 0.0060 | 0.012 | 0.0301 | 0.0361 | 0.0060 | 0.9096 |
| in 3,1,2 | 0.0067 | 0.0133 | 0.04 | 0.02 | 0.0067 | 0.9133 |
| in 3,2,1 | 0.0061 | 0.0122 | 0.0549 | 0.0244 | 0.0061 | 0.8963 |

**Fig. 2.** Probabilities of the Message Ordering from Mixminion Experiments

# Cover & Thomas: Ways to Finding Capacity

- A “gradient climb” algorithm e.g. Frank-Wolfe.
- Kuhn-Tucker Theorem/Lagrange multipliers.
- The Blahut-Arimoto algorithm

# Blahut-Arimoto Algorithm

How do we find the maximising input distribution?

$I(X;Y) = H(X) - H(X|Y)$

$= \sum_{x,y} p(x)\, W(y|x) \log \left( W(y|x) \,/\, \sum_{x'} p(x')W(y|x) \right)$

$= \sum_{x} p(x).D\left( W(\_|x) \;||\; \sum_{x'} p(x')W(\_|x') \right)$

$= \sum_{x} p(x).D_x(W||pW)$

Distribution of y given x

Distribution of y

$\sum_x p(x).D_x(W||pW) \leq C(W) \leq \mathrm{Max}_x\, D_x(W||pW)$

# Blahut-Arimoto Algorithm.

1) Guess an input distribution $p^0(a)$ e.g., uniform

2) Improve the guess, for all x:

$$p^{n+1}(x) = \frac{\exp( D_x(W \,||\, p^nW) )}{\sum_{x'} \exp( D_{x'}(W \,||\, p^nW) )}$$

3) Repeat until $I(p^n,W) - \mathrm{Max}_x D_x(W||p^nW) < e$

Can be tweaked for super linear convergence, conditional mutual information etc.

# Method of Analysing Anonymity

- To analyse a system we define the inputs and outputs.
  - Some abstraction might be needed to make the number of observations manageable
- We run tests of the system for each input.
- From these tests we estimate a matrix.
- We estimate capacity, from the matrix.

# Terms

Everybody else

$W$ : the matrix of the true system.

$\hat{W}_n$ : a matrix estimated from n samples.

$Q$ : the input dist. that maximise M.I.

$\hat{Q}_m(\hat{W}_n)$ : the B.A. algorithm applied to $\hat{W}_n$.

$C = I(Q,W)$ : the true system capacity.

$\hat{C}_{nm} = I(\hat{Q}_m(\hat{W}_n),\hat{W}_n)$ : estimate of capacity ??

Us

# Prob. Observed from Mixminion Node

| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0.0 | 0.0118 | 0.0473 | 0.0118 | 0.0059 | 0.9231 |
| in 1,3,2 | 0.0117 | 0.0 | 0.0351 | 0.0292 | 0.0 | 0.924 |
| in 2,1,3 | 0.005 | 0.0222 | 0.0278 | 0.0444 | 0.0056 | 0.8944 |
| in 2,3,1 | 0.0060 | 0.012 | 0.0301 | 0.0361 | 0.0060 | 0.9096 |
| in 3,1,2 | 0.0067 | 0.0133 | 0.04 | 0.02 | 0.0067 | 0.9133 |
| in 3,2,1 | 0.0061 | 0.0122 | 0.0549 | 0.0244 | 0.0061 | 0.8963 |

**Fig. 2.** Probabilities of the Message Ordering from Mixminion Experiments

# Observation from a running Mixminion Node

| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0.0 | 0.0118 | 0.0473 | 0.0118 | 0.0059 | 0.9231 |
| in 1,3,2 | 0.0117 | 0.0 | 0.0351 | 0.0292 | 0.0 | 0.924 |
| in 2,1,3 | 0.005 | 0.0222 | 0.0278 | 0.0444 | 0.0056 | 0.8944 |
| in 2,3,1 | 0.0060 | 0.012 | 0.0301 | 0.0361 | 0.0060 | 0.9096 |
| in 3,1,2 | 0.0067 | 0.0133 | 0.04 | 0.02 | 0.0067 | 0.9133 |
| in 3,2,1 | 0.0061 | 0.0122 | 0.0549 | 0.0244 | 0.0061 | 0.8963 |

**Fig. 2.** Probabilities of the Message Ordering from Mixminion Experiments

Leakage = 0.0249 bits

# Convergence

Theorem: $\hat{C}_{n,m}$ almost surely convergences to C as $n,m \rightarrow \infty$

i.e., for any $p_e$ and error e there exists n‘ & m’ such that for n > n’ and m > m’:

$$p(| C - \hat{C}_{n,m} | > e ) < p_e$$

# The Distribution of Anonymity

We can get bounds on the error by ask what distribution $\hat{C}_{n,m}$ comes from.

Adapting a statistical method from Rao:

- We find the Taylor expansion of the $\hat{C}_{n,m}$
- We drop the terms smaller than $\text{sampleSize}^{-2}$
- We then calculate the mean and variance.
- We find the distribution using the central limit theory.

# Estimated Value

As we can't find the distribution for the maximising distribution we relate our estimate to $I(\hat{Q}_m(\hat{W}_n),W)$

Lemma: The estimate

- is less than or equal to the capacity,
- equals zero if, and only if, the capacity equals zero.

# Expectation and Variance

To find a distribution we need to find the expectation:

E(X): the average value

And the variance:

$$\text{Var}(X) = E(\text{mean} - x)^2$$

# What We Know

$K_{ij}$ is the number of times the pair (i,j) shows up in our test.

Let the true prob: $p(i,j) = {}^{h}K_{ij}/n$

Then maximum likehood tells us that

- $E(K_{ij} - {}^{h}K_{ij}) = 0$
- $E((K_{ij} - {}^{h}K_{ij})^2) = p(i).\ W(j|i)(1-W(j|i))$
- $E((K_{ij} - {}^{h}K_{ij})^3) = K_{ij}(2W(j|i)^2-3W(j|i)+1)$ ...

# Taylor's Theorem

To find the value of a function at x (near a):

$$f(x) = f(a) + \frac{f'(a)(x-a)}{1!} + \frac{f''(a)(x-a)^2}{2!} + \frac{f''(a)(x-a)^3}{3!} + \ldots$$

We take I(X,_) as “f”, $W_n$ as “x” and W as “a” to give get an expression for the estimate in terms of the true value.

# Taylor Expansion of Entropy

$$I_n(X,Y) = H(X) + H_n(Y) - H_n(X,Y)$$
$$E(I_n(X,Y)) = E(H(X)) + E(H_n(Y)) - E(H_n(X,Y))$$

$$H(X,Y) = -\sum_{x,y} p(x,y)\log(p(x,y))$$
$$H_n(X,Y) = -\sum_{x,y} K_{ij}/n.\log(K_{ij}/n)$$
$$H_n(X,Y) = -\sum_{x,y} {}^hK_{ij}/n - 1/n.\sum_{x,y}(1+ {}^hK_{ij}/n)$$
$$-\sum_{x,y}(K_{ij}-{}^hK_{ij})^2/n.\ {}^hK_{ij})$$
$$+ \sum_{x,y}(K_{ij}+ {}^hK_{ij})^3/6n.{}^hK_{ij}{}^2) + O(n^{-2})$$

$$E(H_n(X,Y)) = H(X,Y) - I(J-1)/2n + O(n^{-2})$$

# For Non-Zero Mutual Information

When the true value is not 0, an estimation of capacity is drawn from a normal distribution with:

Mean: $I(\hat{Q}_m(\hat{W}_n),W) + \frac{(I-1)(J-1)}{2n} + O(n^{-2})$

I = no. of Inputs, J = no. of Outputs

Variance: ...

# Variance

$$\frac{1}{n} \sum_x Q(x).\left(\sum_y W(y|x). \left(\log\left(\frac{Q(x).W(y|x)}{\sum_{x'} Q(x')W(y|x')}\right)\right)^2\right)$$
$$- \left(\sum_y W(y|x). \log\left(\frac{Q(x).W(y|x)}{\sum_{x'} Q(x')W(y|x')}\right)\right)^2$$
$$+O(n^{-2})$$

# When I = 0

- The $O(n^{-1})$ term disappears with X and Y are independent.
- In which case we need to find the $O(n^{-2})$ term.
- Following Rao, we observe when I = 0 :

$$\sum_{ij} \left( \frac{(K_{ij} - E(K_{ij}))^2}{E(K_{ij})} \right) \sim \chi^2$$

and that this approximates mutual information.

# Results for I = 0

When the true value is 0, an estimation of capacity (or mutual information) is drawn from the distribution:

$$2n.I \sim \chi^2((noOfInputs-1)(noOfOutputs-1))$$

Mean: $(noOfInputs-1)(noOfOutputs-1)/2$

Variance: $(noOfInputs-1)(noOfOutputs-1)/2n^2$

# Upper Bound on the Variance

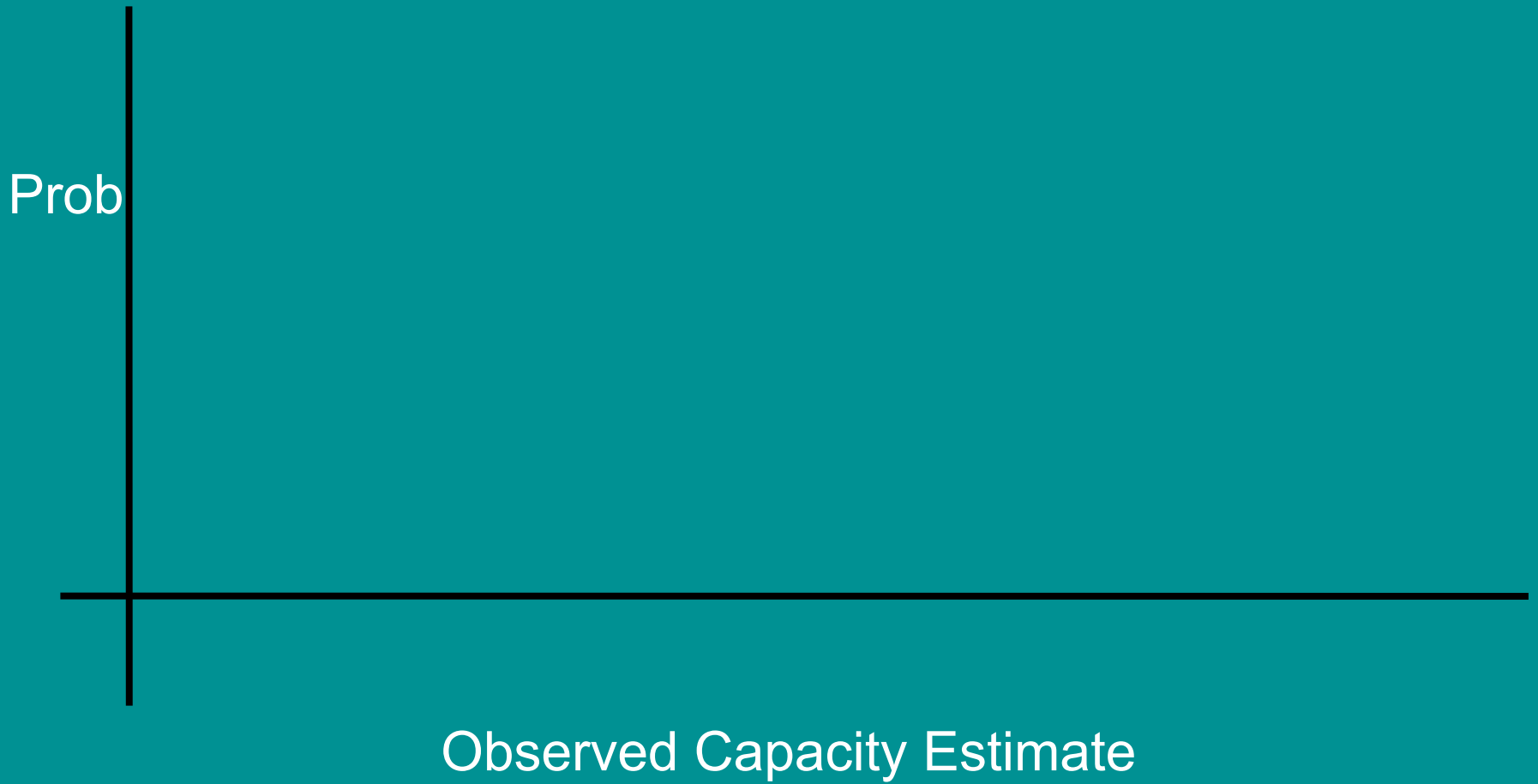
- In both cases var(C(W)) < I.J / n
- Rule of thumb:
  - If I.J >> n the variance will be low and the results actuate.
  - If you can get this many samples then statically analysis is useful, otherwise not.

# To Analyse a System.

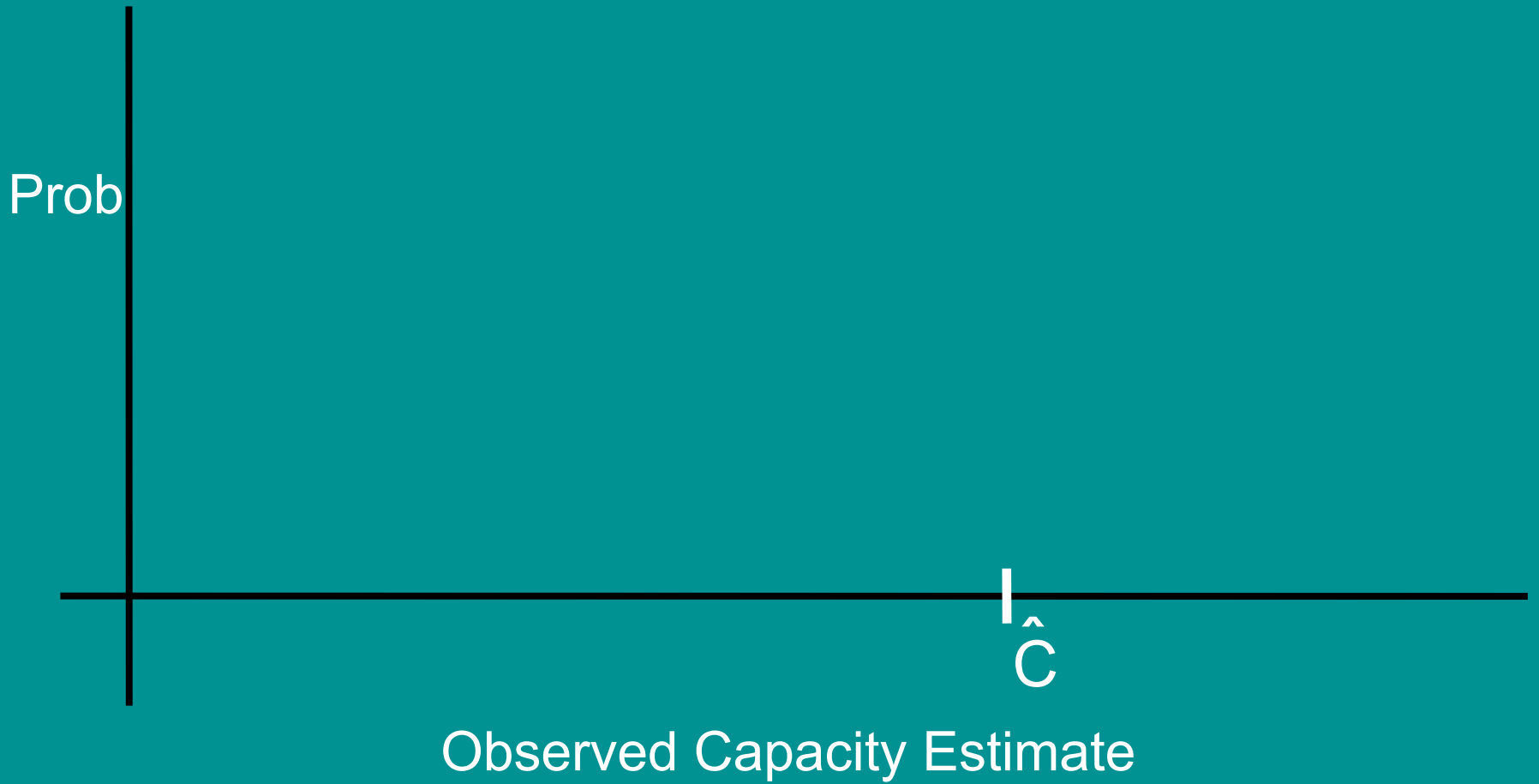
- We define the inputs (I) and outputs (J).
- Run n tests of the system with n >> I.J
- Estimate the matrix and find $\hat{C} = I(\hat{Q}_m(\hat{W}_n), \hat{W}_n)$
- Point Estimate is:

$$\text{Max}\left(0, I(\hat{Q}_m(\hat{W}_n), \hat{W}_n) - (I-1)(J-1)/2n\right)$$

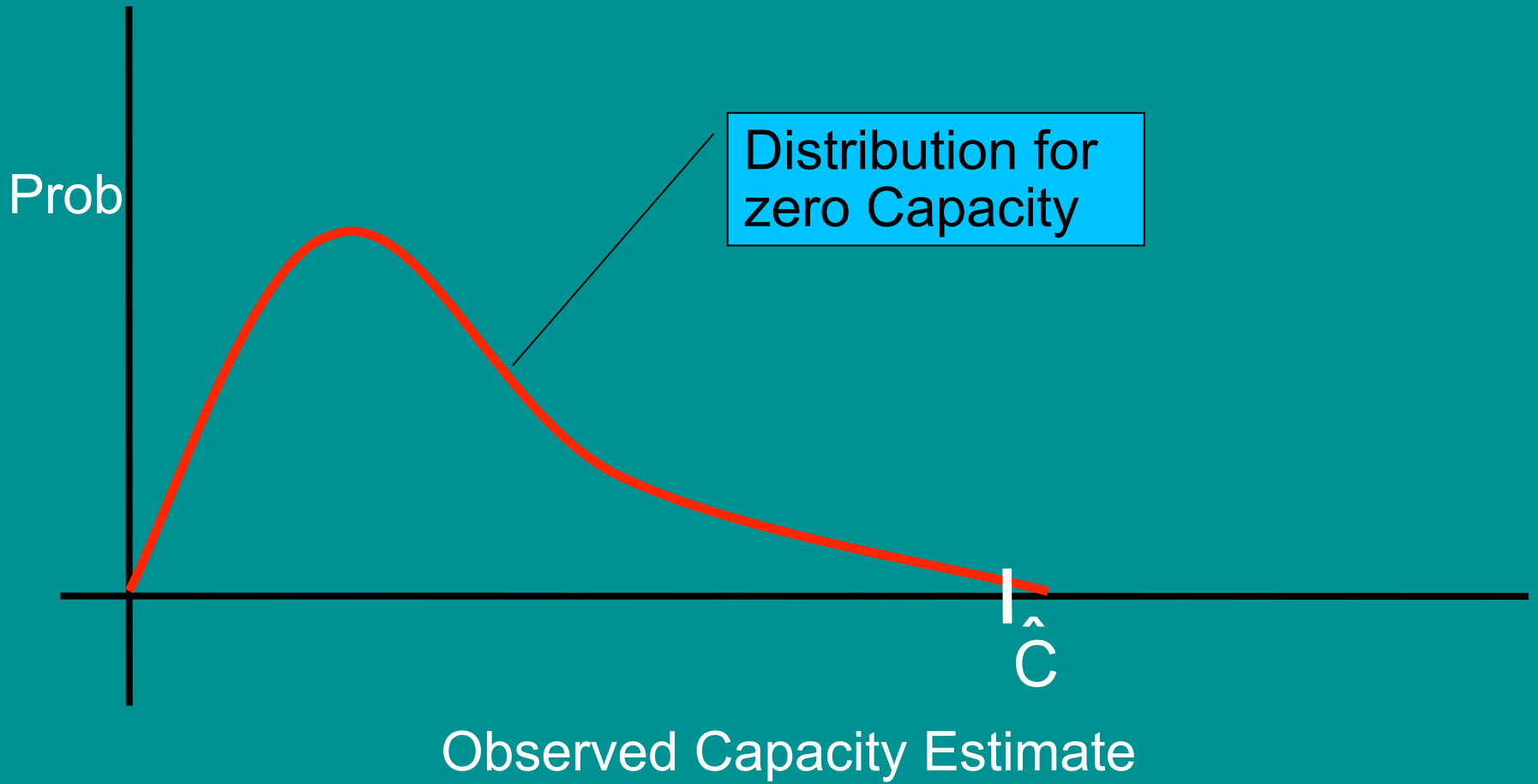
# Using the Distributions



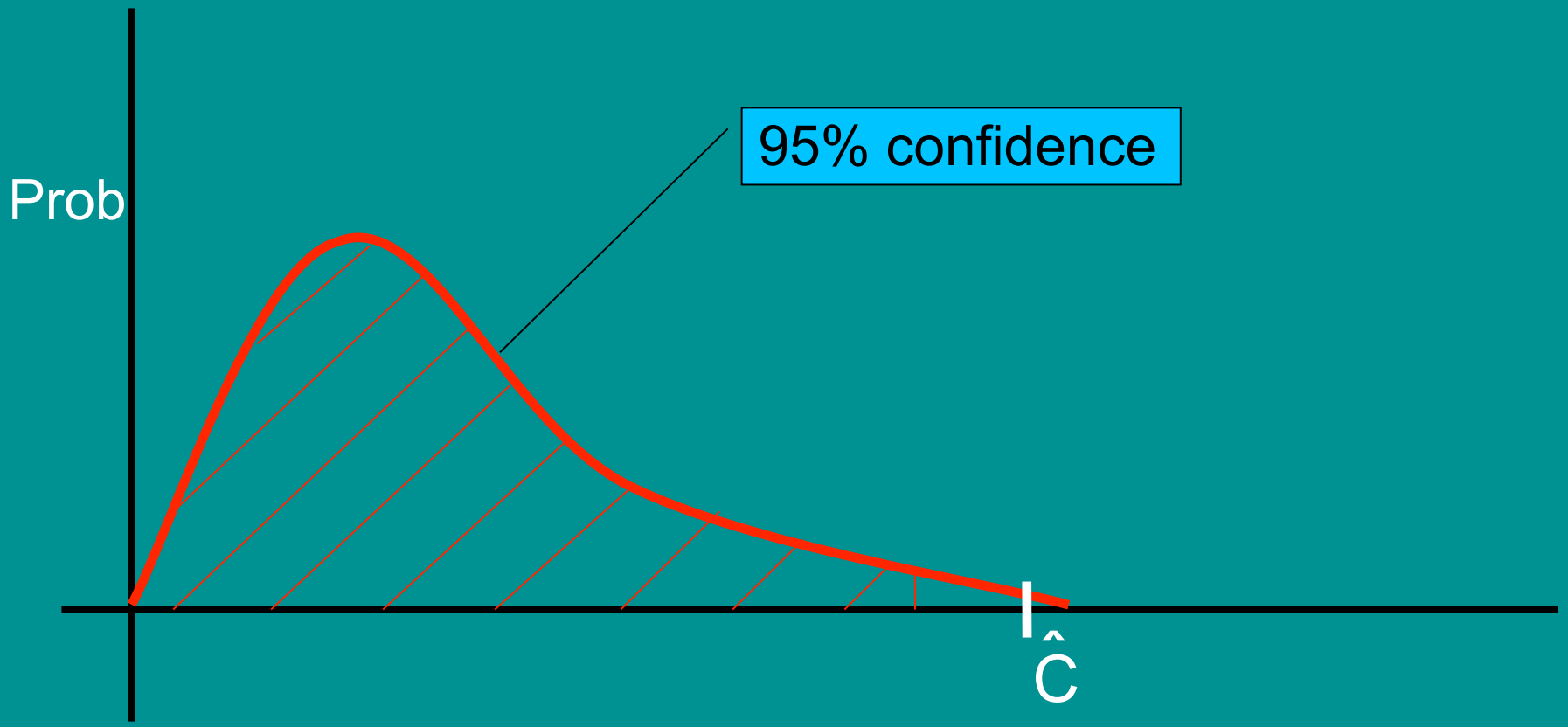
# Using the Distributions



# Using the Distributions

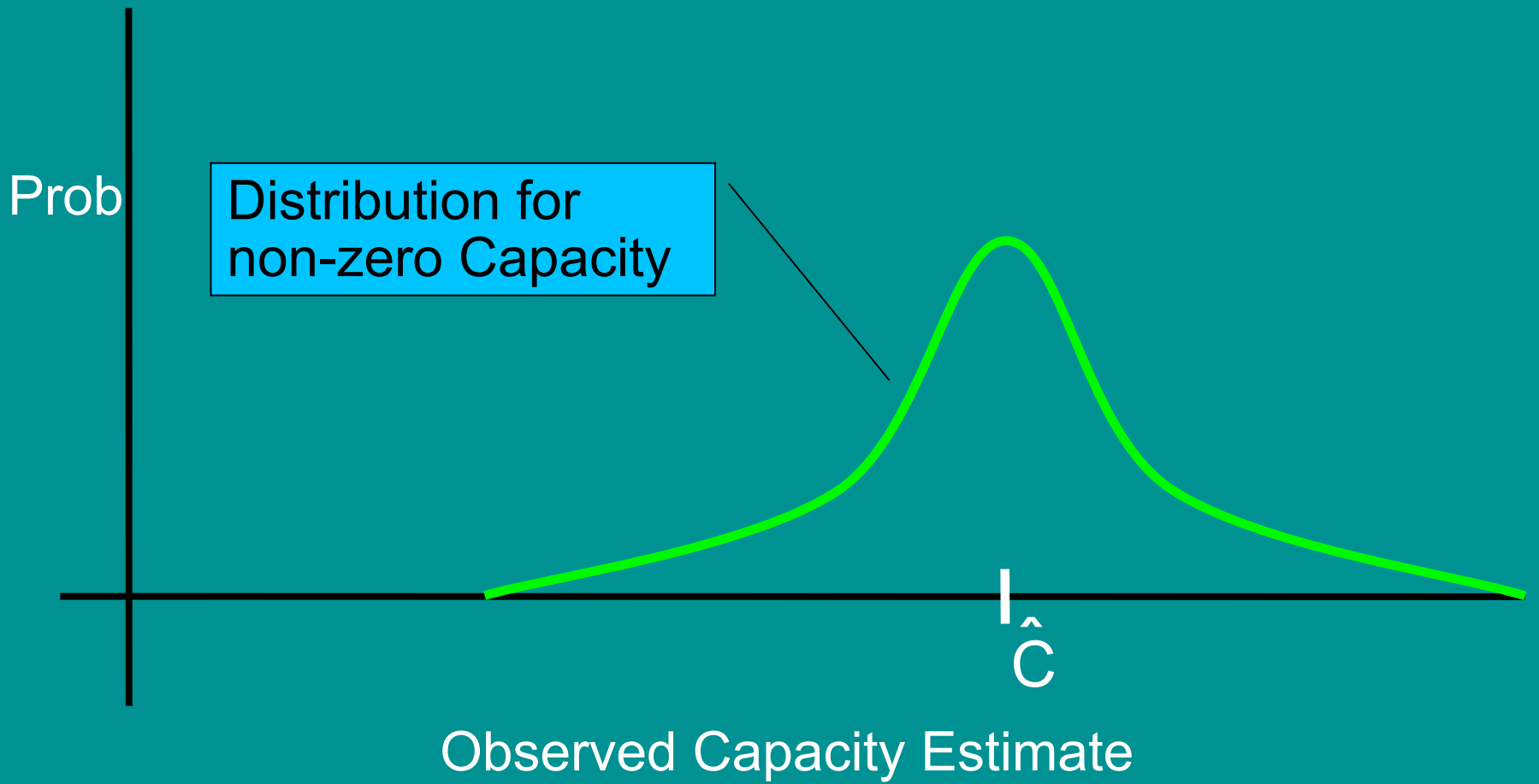


# Using the Distributions

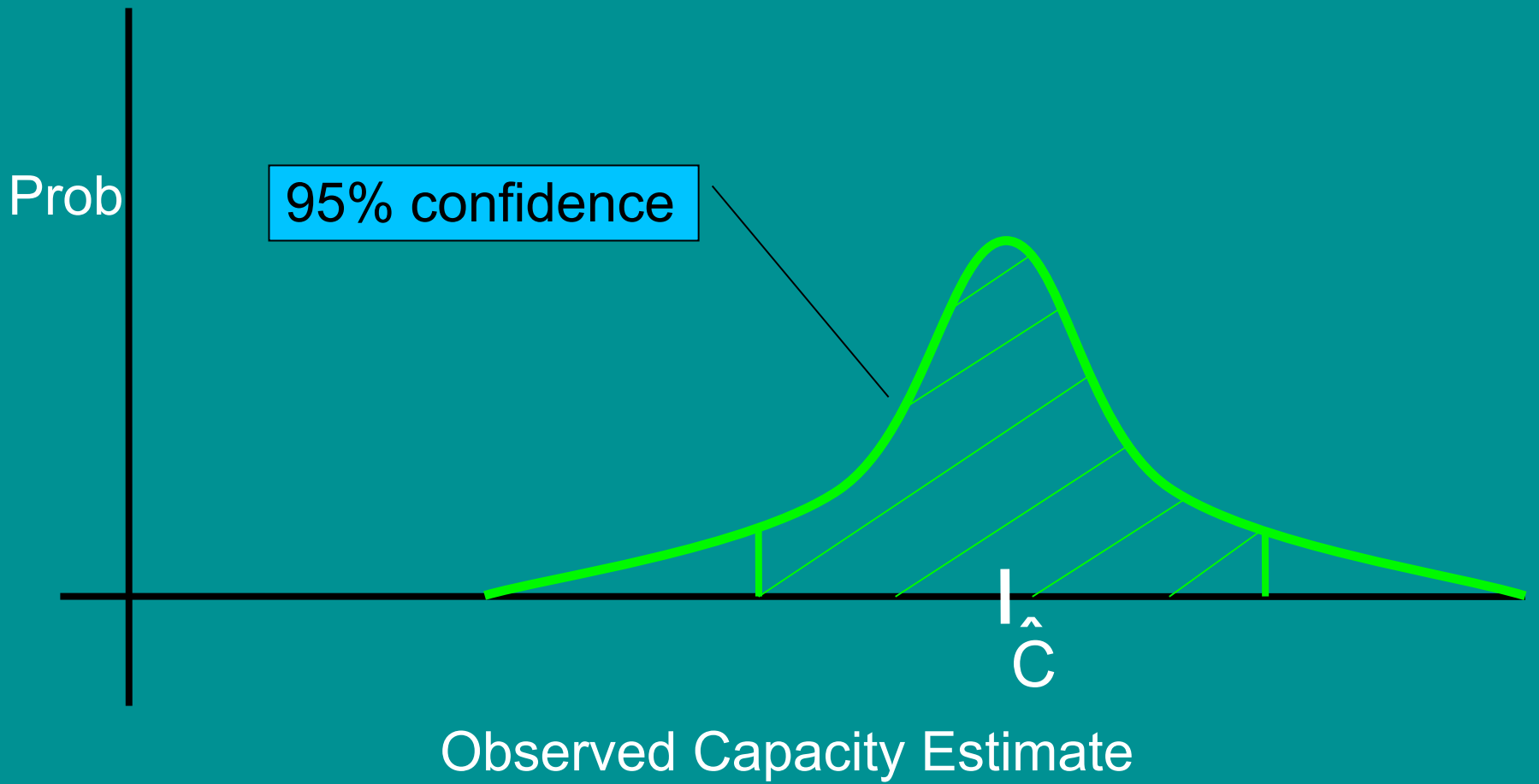



Observed Capacity Estimate

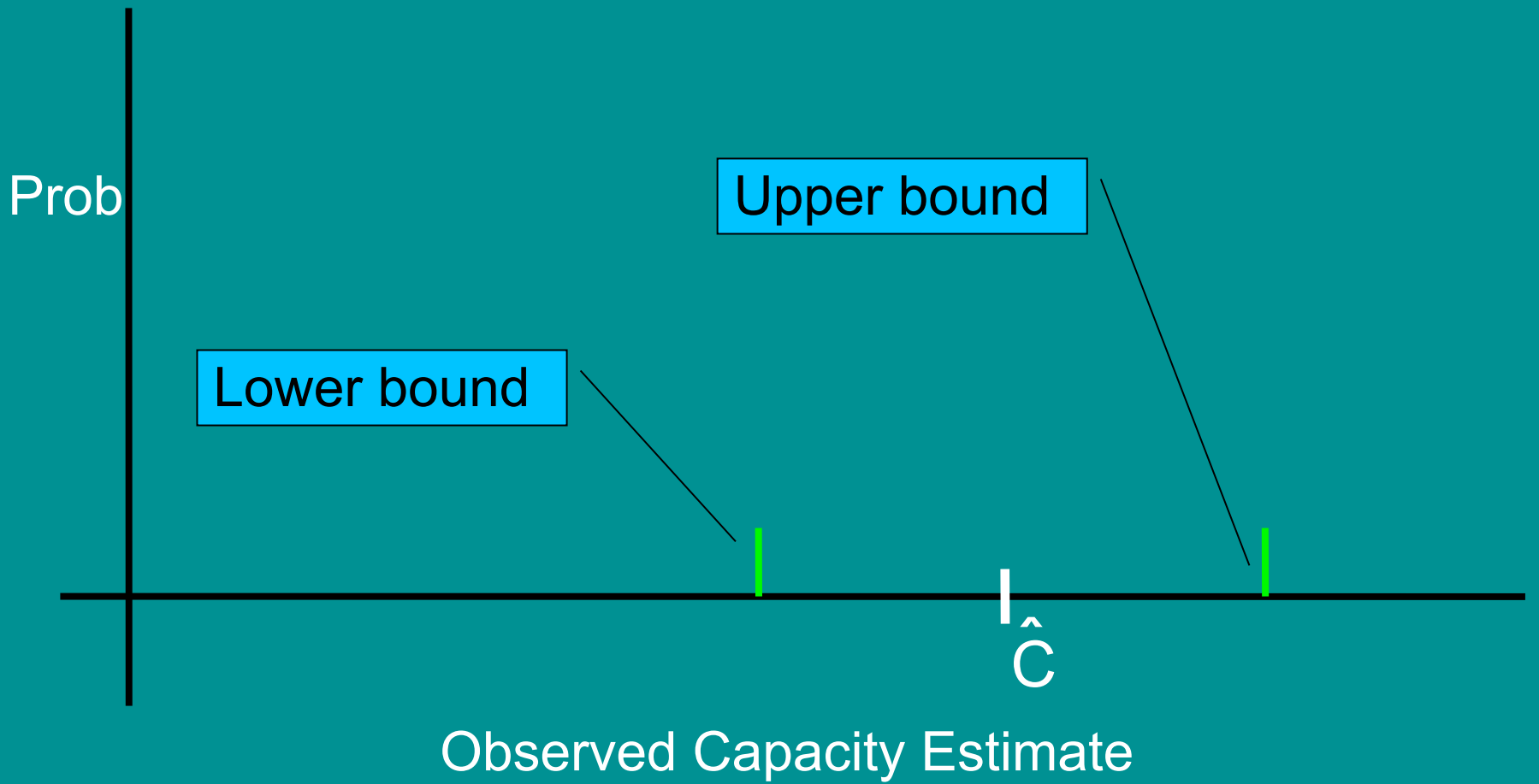
# Using the Distributions



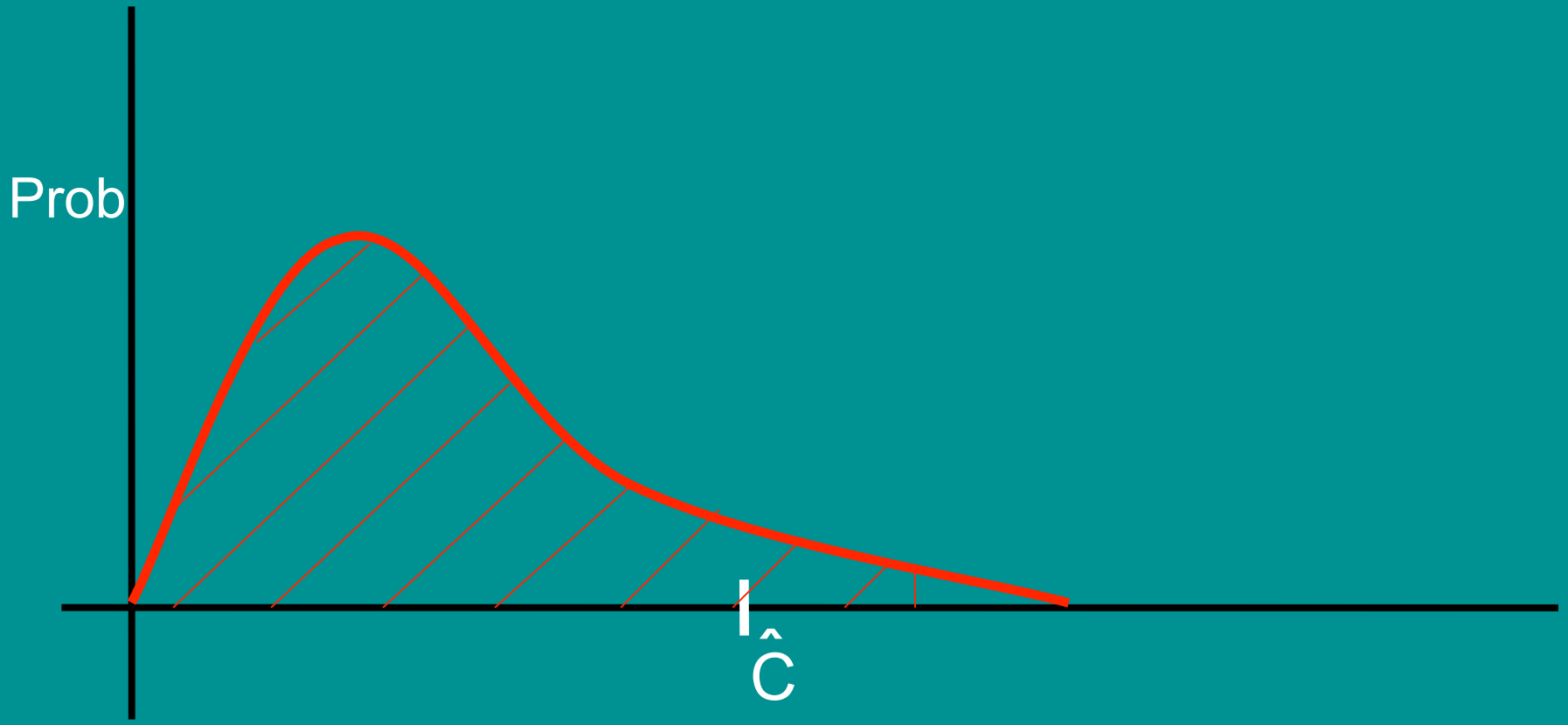
# Using the Distributions



# Using the Distributions

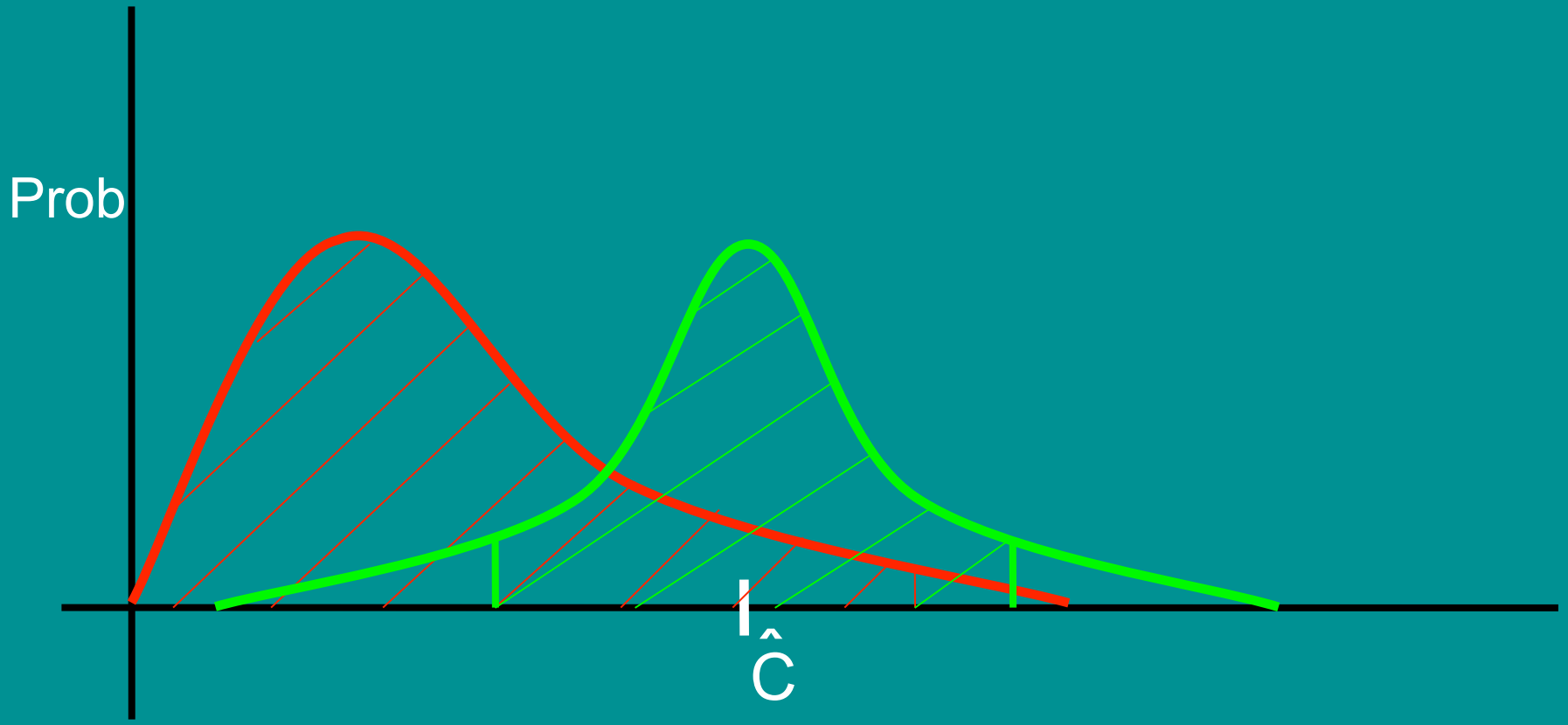


# Using the Distributions

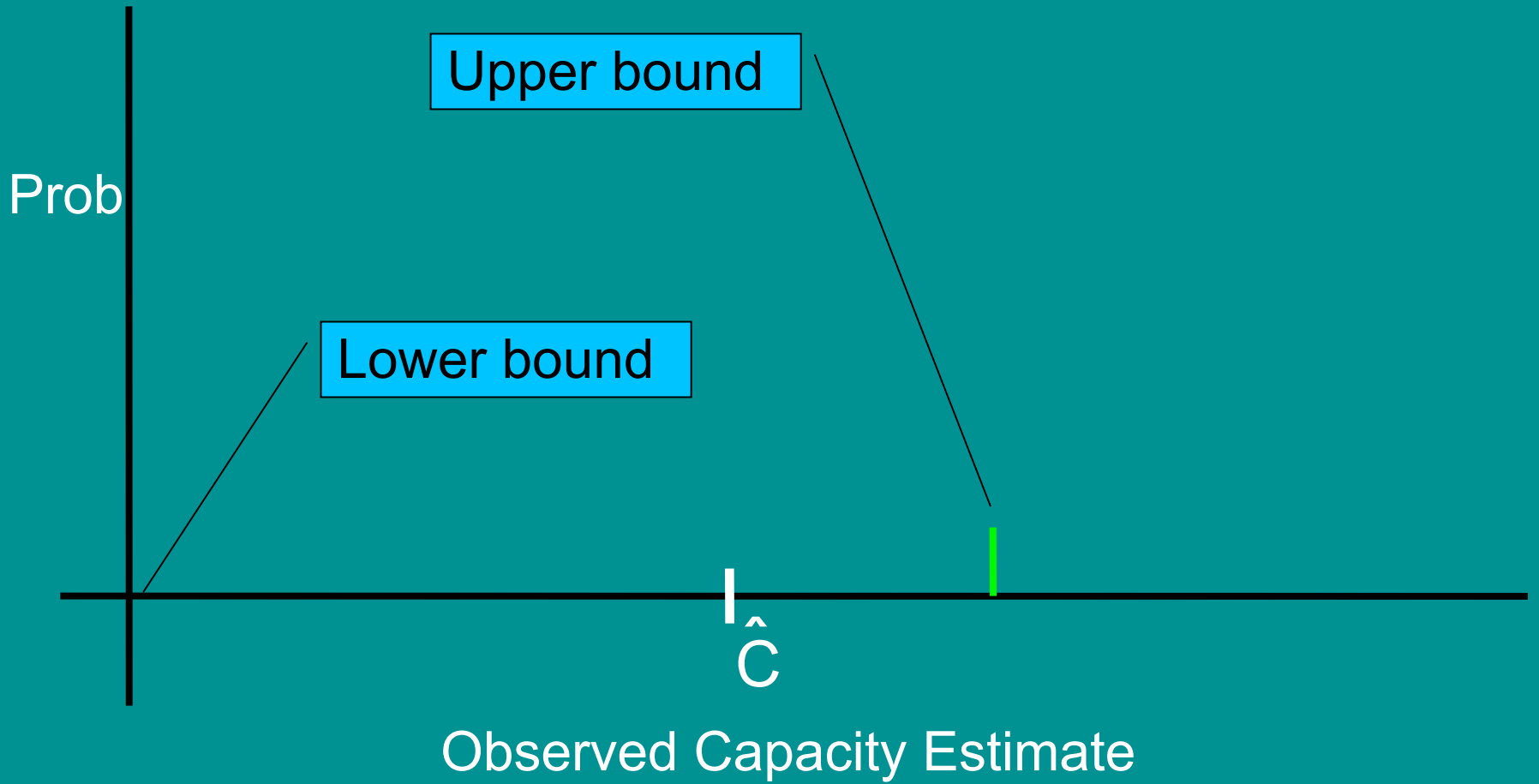



Observed Capacity Estimate

# Using the Distributions



# Using the Distributions



# Test for Zero Leakage

- But what if we want to know if the leakage is really zero?
- What distinguishes the zero from the non-zero case is the variance:
  - $O(n^{-1})$ for non zero
  - $O(n^{-2})$ for zero.
- A large enough sample size will always tell these apart, with a given certainty.

# Test for Zero Leakage

- Run 40 tests of the system and calculate the observed variance “o” in the tests results.
- Test o against the predicated variance for zero and non-zero observations.
- If it matches the zero predication but not the non-zero we can conclude that there is zero leakage.
- If it only matches the non-zero predication then we can find the confidence interval for the results.
- If it matches both then increase the sample size.

# Prob. Observed from Mixminion Node

| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0.0 | 0.0118 | 0.0473 | 0.0118 | 0.0059 | 0.9231 |
| in 1,3,2 | 0.0117 | 0.0 | 0.0351 | 0.0292 | 0.0 | 0.924 |
| in 2,1,3 | 0.005 | 0.0222 | 0.0278 | 0.0444 | 0.0056 | 0.8944 |
| in 2,3,1 | 0.0060 | 0.012 | 0.0301 | 0.0361 | 0.0060 | 0.9096 |
| in 3,1,2 | 0.0067 | 0.0133 | 0.04 | 0.02 | 0.0067 | 0.9133 |
| in 3,2,1 | 0.0061 | 0.0122 | 0.0549 | 0.0244 | 0.0061 | 0.8963 |

**Fig. 2.** Probabilities of the Message Ordering from Mixminion Experiments

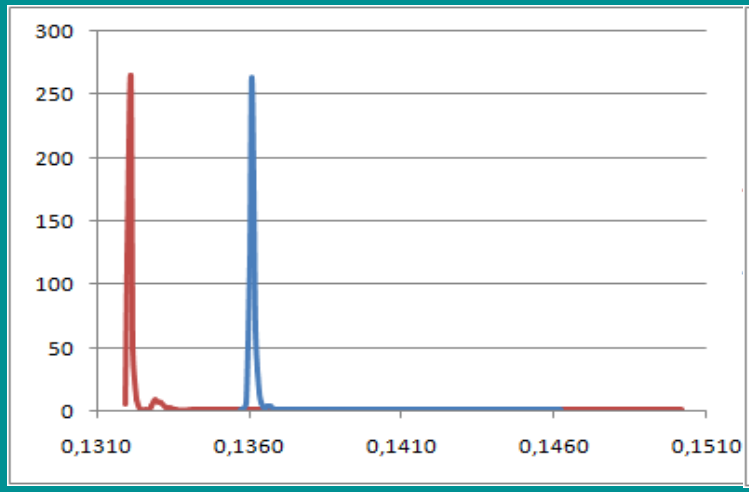
# Observation from a running Mixminion Node

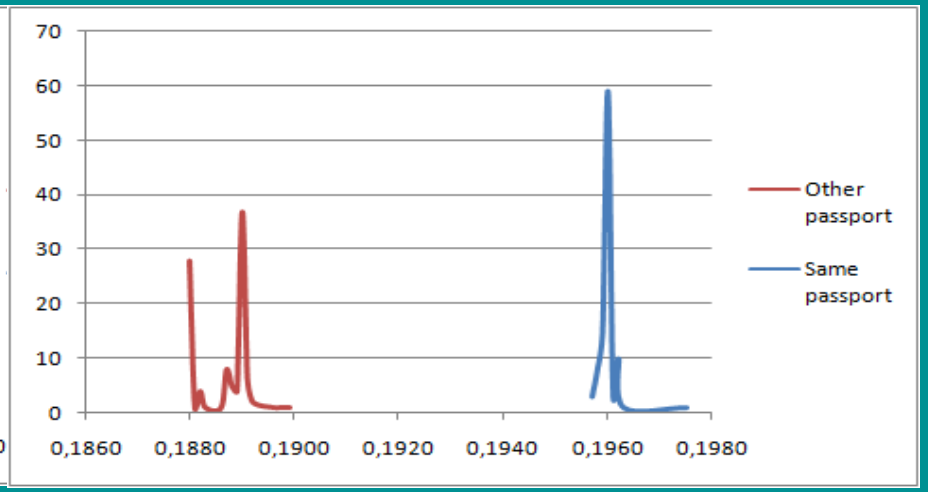
| Message orderings | out A,B,C | out A,C,B | out B,A,C | out B,C,A | out C,A,B | out C,B,A |
|---|---|---|---|---|---|---|
| in 1,2,3 | 0.0 | 0.0118 | 0.0473 | 0.0118 | 0.0059 | 0.9231 |
| in 1,3,2 | 0.0117 | 0.0 | 0.0351 | 0.0292 | 0.0 | 0.924 |
| in 2,1,3 | 0.005 | 0.0222 | 0.0278 | 0.0444 | 0.0056 | 0.8944 |
| in 2,3,1 | 0.0060 | 0.012 | 0.0301 | 0.0361 | 0.0060 | 0.9096 |
| in 3,1,2 | 0.0067 | 0.0133 | 0.04 | 0.02 | 0.0067 | 0.9133 |
| in 3,2,1 | 0.0061 | 0.0122 | 0.0549 | 0.0244 | 0.0061 | 0.8963 |

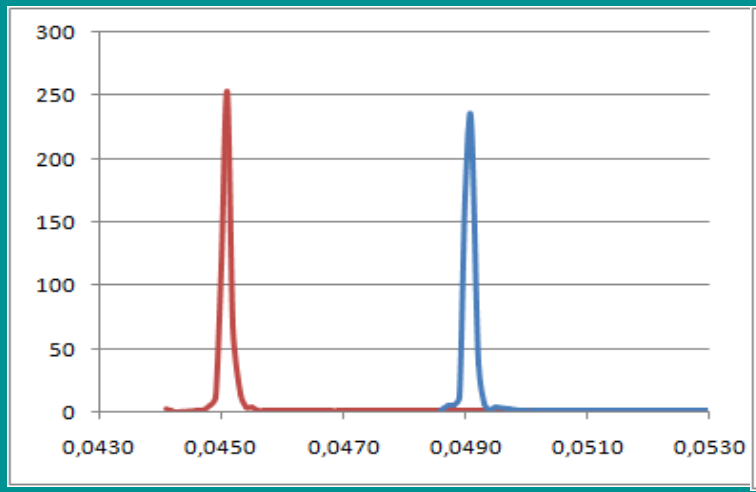
**Fig. 2.** Probabilities of the Message Ordering from Mixminion Experiments

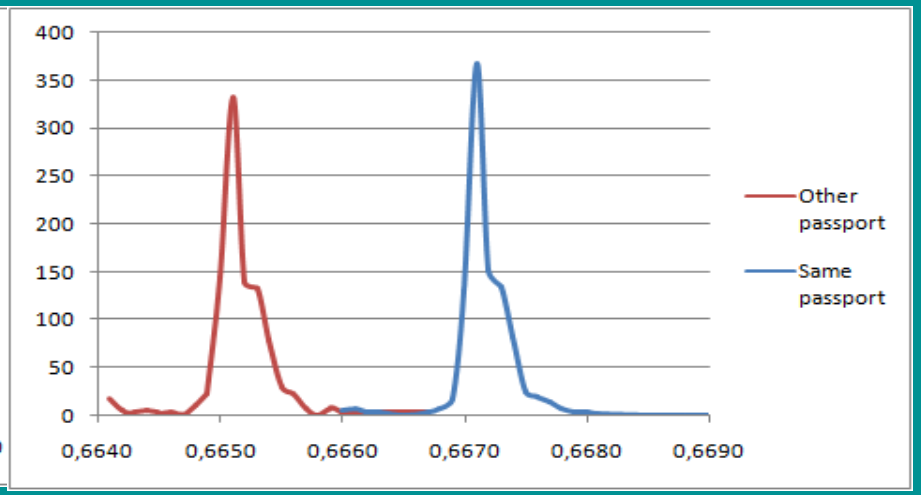
Leakage = 0.0249 bits
Confidence interval for zero leakage = 0, 0.0355

# Back to e-Passports:



# Information-theoretic Measurement of Information Leakage from Passports

Estimated Leakage in bits, 95% confidence:

- UK : 0.9517
- German : 0.9716
- Greek : 0.9921
- Russian : 1

As there is only one “input” these equate to the probability of a successful guess.

# Conclusion

- Information leaks are often due to the implementation.
- We can estimate information leakage statistically from trail runs of a real system.
- This may find errors that model checking would miss.
- State-space doesn’t matter.

# Further Work

- Proper treatment of continuous data.
- Apply to other forms of information theoretic measurement.
- Better ways to apply this to real systems.

Questions?