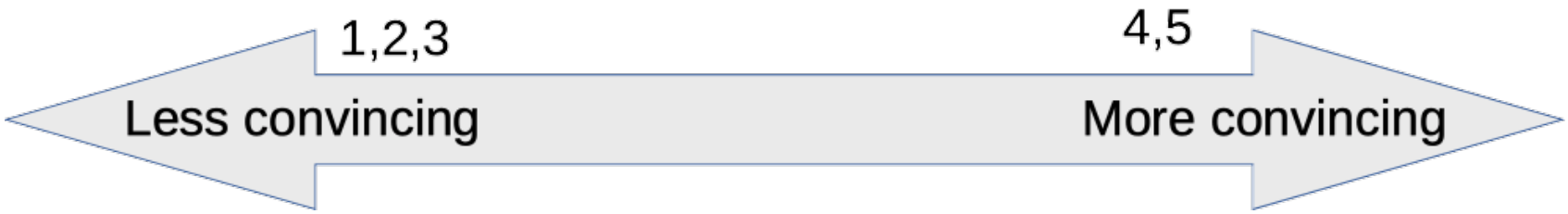
Modeling Entropy Sources

John Kelsey, NIST and KU Leuven

Overview

- An entropy source submission must include:
  - An explanation of where the entropy comes from
  - An entropy estimate
  - Justification for the entropy estimate
  - Statement about whether source claims iid
- Black-box tests run to estimate entopy from outputs
  - Entropy estimate never larger than submitter estimate
  - Useful as a sanity check
  - Not enough to be convincing



- Entropy estimates and justifications range from extremely sketchy to extremely convincing.

Less convincing:

1. “There’s gotta be a bit in there somewhere”
2. “I ran some statistical tests on it and the outputs passed”
3. ”The 90B tests assessed it at 0.7 bits / output”
4. “This model is based on an extensive literature on the subject”
5. “We took these measurements and ran these experiments to verify our model”

More convincing

Preliminaries

- Requieremements from 90B
- Questions to ask
- What's an entropy estimate?
- Working backwards
- Physical vs non-physical sources

Requirements on the Noise Source

From Section 3.2.2

- **The operation of the noise source shall be documented**; this documentation shall include a description of how the noise source works, where the unpredictability comes from, and rationale for why the noise source provides acceptable entropy output.
- **Documentation shall provide an explicit statement of the expected entropy provided** by the noise source outputs and provide a technical argument for why the noise source can support that entropy rate.

Questions about the noise source

- **How does the noise source work? (What's unpredictable about it?)**

  "The operation of the noise source shall be documented…"

- **Where does the unpredictability come from?**

  "…where the unpredictability comes from"

- **How much entropy / output is produced?**

  "Documentation shall provide an explicit statement of the expected entropy provided…"

- **How do you know? (Justify the entropy estimate.)**

  "…provide a technical argument for why the noise source can support that entropy rate."

How does the noise source work?

"The operation of the noise source shall be documented…"

- Detailed description
- Diagrams
- Internal measurements of parameters
- Analysis of its behavior

Why is it nondeterministic?

"…where the unpredictability comes from"

- What about the noise source is not deterministic?
- Where does that nondeterministic behavior come from?
- What prevents an attacker from predicting the behavior of the source?

How much entropy / output is produced?

> *"Documentation shall provide an explicit statement of the expected entropy provided…"*

- Requires knowing something about probability distribution on outputs
- Need to estimate or upper bound P[max]
- H[min] = -lg(P[max])
  - Higher P[max] → lower H[min]

P[max]	H[min]
0.50	1.00
0.55	0.86
0.60	0.74
0.70	0.51
0.80	0.32
0.90	0.15
0.95	0.07

Justify the entropy estimate

> *"…provide a technical argument for why the noise source can support that entropy rate."*

- Probability model of some kind for outputs
- Model for nondeterministic process that produces outputs
- Justification and evidence for models

Statistical tests alone are NOT strong evidence of entropy estimate!

What is an entropy estimate?

- Ultimately, an entropy estimate is an estimate of P[max].

H[min] = -lg(P[max])

P[max] =

Maximum for all possible outputs x

Pr[output = x | all attacker knowledge] ← 90B

Usually upper-bound P[max] to deal with: → *P*[max] >= P[max]*

Intuition: P[max] in 90B

Assume:

- Attacker has a very good understanding of your source
- He's examined millions of samples from this particular device
- He's seen all previous samples since startup

- Attacker wants to predict next output
- How do we bound his probability of success?

$$P[max] = Max\ (all\ x)\ (\ Pr[\ x = output \mid all\ attacker\ information\ ]\ )$$

Intuition: Working backwards

- What we need: upper bound on P[max]
  - Gives lower bound on H[min]

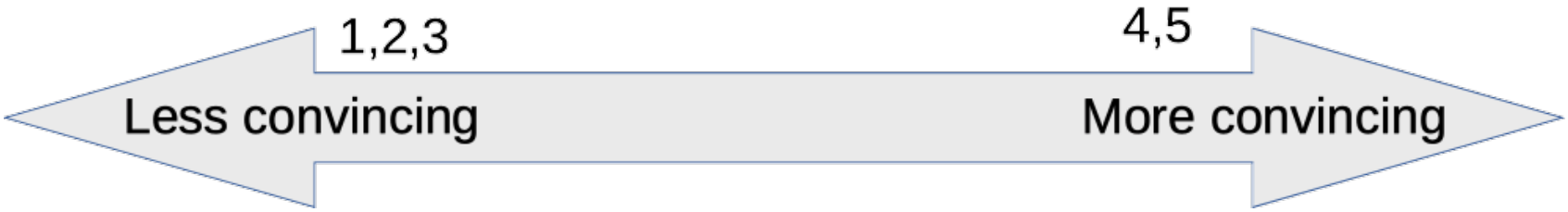
- To compute bound, we need probability model on outputs
  - Doesn't need to specify everything, but needs to let us bound P[max]
  - Must agree with observed properties of outputs
  - Must make sense in light of operation of noise source

- To construct probability model, we may need to model noise source
  - What's the unpredictable part?
  - Probability distribution / model for unpredictability
  - How that affects outputs

Physical vs non-physical sources

- What we need: A probability model for the output.
  - So we can bound P[max]
- How we can get that: physical sources
  - Build a model of the source's behavior
  - Estimate the parameters of that model
  - Use that model to produce the probability model
- How we get that: nonphysical or "found" sources
  - Try to model source's behavior
  - Justify some claim about probability model
  - Nonphysical sources are typically way too complex to model well.

What do we want from justification?



- Suppose you're going to trust this source with something important
- What kind of justification would you find convincing?
- What kind would you find worrying?

Designing The Source

- Designing source and model together
- Designing with testing in mind
- Thinking about failures

Designing the source

- Design the source with this process in mind from beginning!

- Many problems can be headed off in the design
  - Access to raw bits for validation and health testing
  - Designing the source to simplify the modeling
  - Building in mechanisms to detect or prevent failures internally

- Complexity is NOT your friend
  - Super complex designs are hard to test, validate, and verify

Easy to model

- Design the source with the model in mind
- Model should be simple enough to be tractable
  - Nice if you can find related stuff in the literature
- Ensure parameters can be measured/estimated
  - Sometimes parameters can be designed in or set in the field!
  - Other times they can be measured externally or looked up
- Ensure model can be checked
  - Access to raw outputs of individual components helps

Easy to test

The noise source should be designed to be easy to test

- Defined mechanism for getting the raw bits out for validation testing
  - Can be disabled when it's shipped
- Sometimes need access to other raw internal values to test model
- May need to be able to disable or turn off some parts for testing
- Justification for why this gives same raw bits used internally
- Defined access to raw bits for health testing
  - Health tests must have access to raw bits

Thinking about failures

How can the source fail?

- Total failures = things go catastrophically wrong
- Model failures = parameters of model are wrong
  - So entropy estimate isn't right

During design, think about how it can fail:

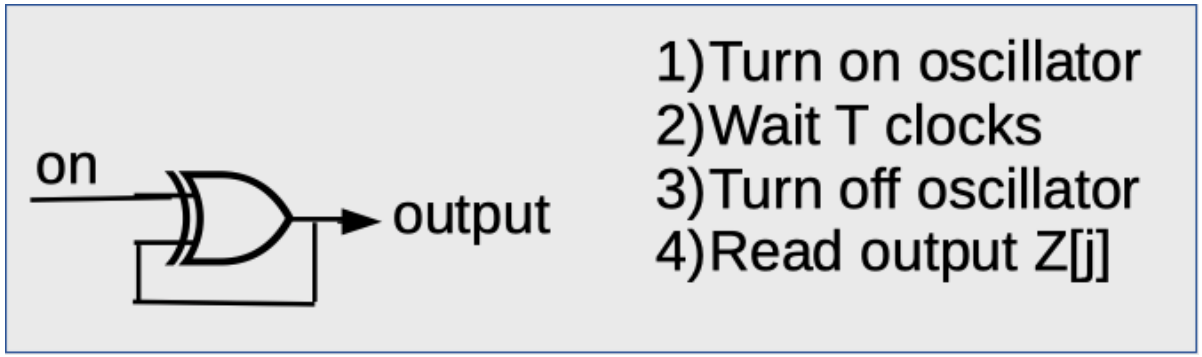
- Can you change design to prevent the failure?
- Can you detect failure?
  - New health test?
  - Internal measurements during operation?

Getting to an entropy estimate

1. Description of noise source operation
2. Explanation of where unpredictability comes from
3. Model of noise source's behavior
4. Model of probability distribution of output
5. Estimate or upper-bound on P[max] ← This is what we need!

How will you get entropy estimate?
How will you justify entropy estimate?
How would you know if you were wrong?

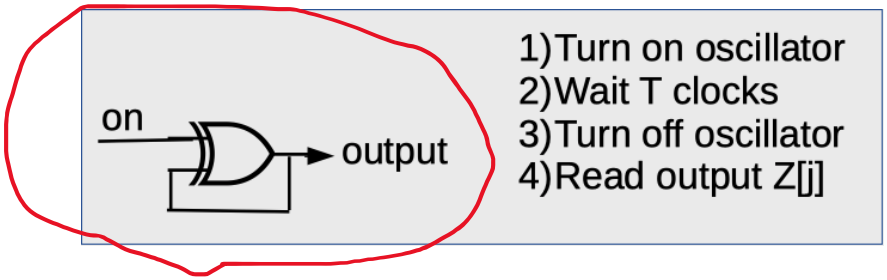
A Quickly Sketched Example



- Let's sketch out a noise source so we can talk about modeling
- Disclaimers:
  - This is a quickly sketched example
  - I'm not an EE
  - I'm not trying to design your noise source
  - This is just an illustration

What kind of justification would you find convincing?

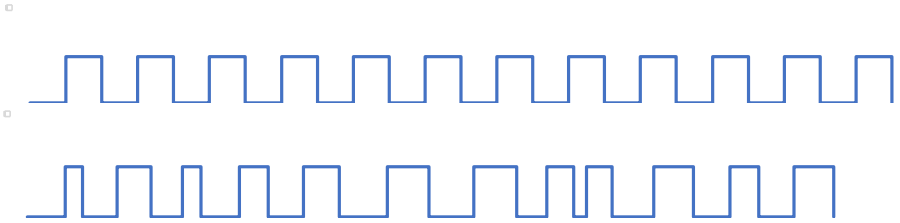
What's going on here?



- When on = 1, this is an unstable oscillator
- When on = 0, this retains its value

Unstable oscillator:

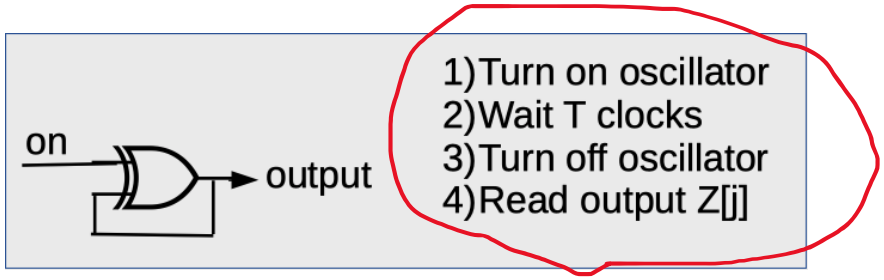
- Keeps transitioning between 0 → 1 → 0 → 1 as long as it's on.
- Time taken to transition varies randomly by a little bit



Stable clock signal

Unstable signal from RO
[VERY EXAGGERATED!]

What about here?



Generating a bit takes T+3 clocks

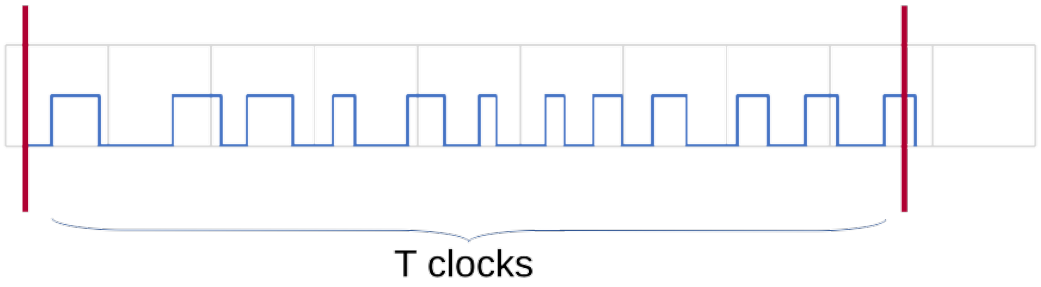
- T is a tunable parameter—designer sets it to get desired properties
- Outputs: Z[1],Z[2],…

- If oscillator changes state EVEN number of times:

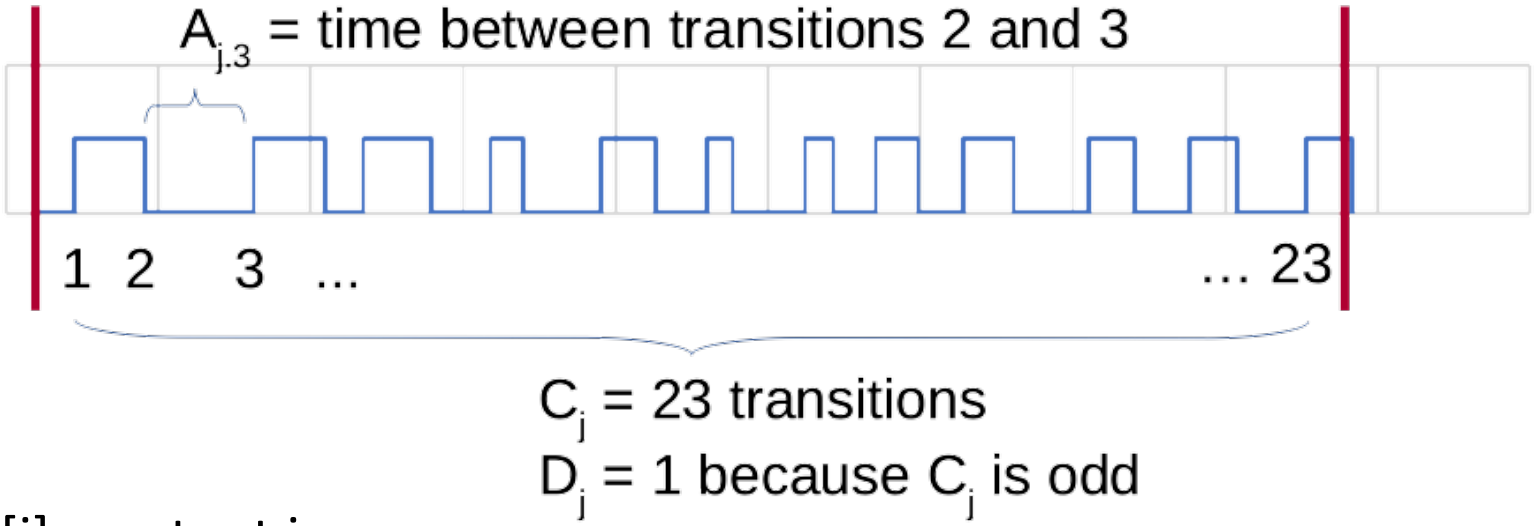
  Z[j] = Z[j-1]

- ODD number of times:

  Z[j] = Z[j-1] XOR 1

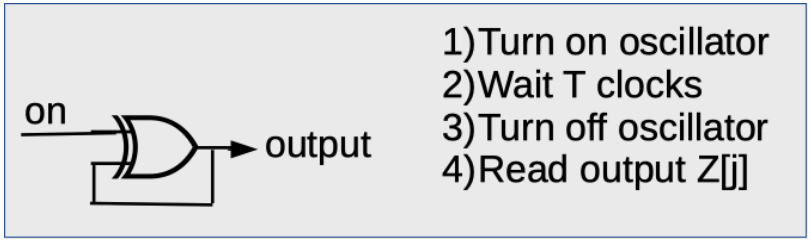


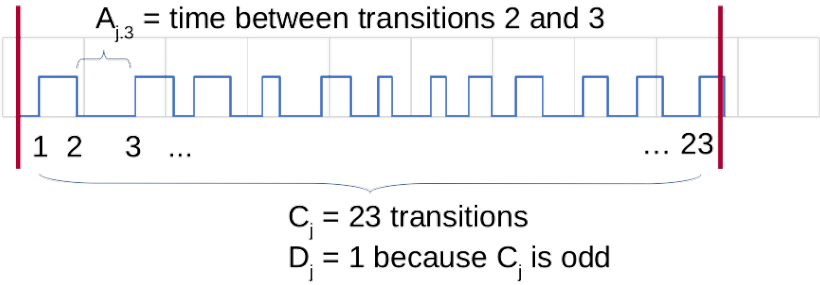
Parameters for Modeling



- Z[j] = output j
- D[j] = parity of # of transitions in output j ← Z[j] = Z[j-1] XOR D[j]
- C[j] = number of transitions in output j ← D[j] = C[j] mod 2
- A[j,k] = time taken for kth transition during jth output

What are the raw bits?

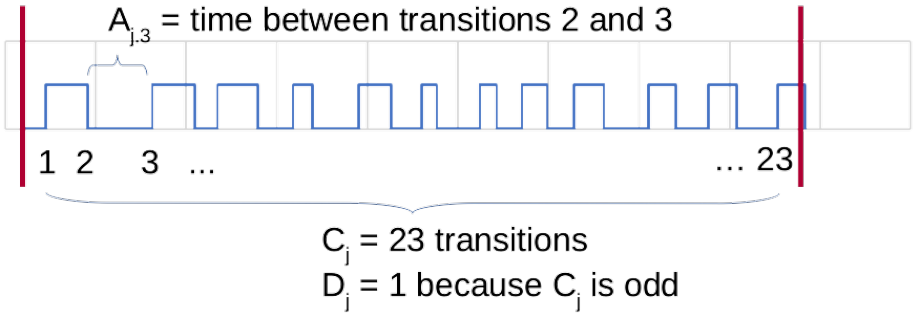




- 90B requires testing the *raw data* from the noise source
- This shows how this notion can be a little confusing
- For this source, entropy comes from whether C[j] is even or odd
- D[j] = C[j] mod 2
- Z[j] = Z[j-1] XOR D[j] ← Entropy in output comes from D[j]

- It probably makes more sense to think of D[j] as the raw data

Modeling D[j]



- The only thing that affects output bits is whether oscillator changes outputs an EVEN or ODD number of times

  EVEN: D[j] = 0

  ODD: D[j] = 1

- We can observe

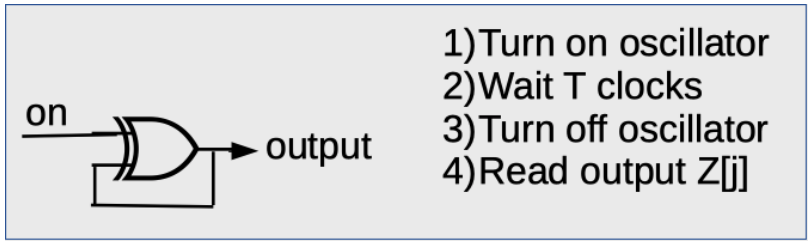
  Pr[D[j] = 1] ← We can also experiment with different T

- We have to use a model to figure out

  Pr[D[j] = prediction(all attacker info)]

Question: Is there some clever way for attacker to predict these bits?

First cut



- Experiment: We can try different values of T
- Increment until we find value T[c] for which D[j] appear unbiased random
- Eventually set T = T[c]*2 as a safety margin

- Model: D[j] are approximately uniform and unbiased
- Entropy estimate based on this
- Iid or non-iid tests will lower the estimate b/c of confidence interval

Good news/bad news

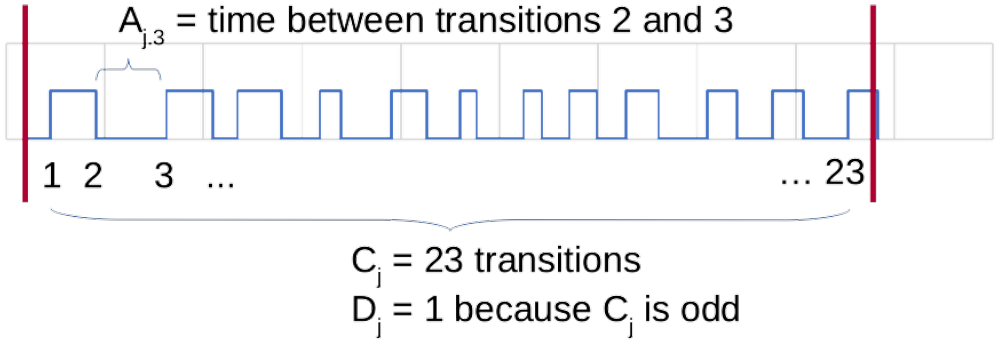
Good news

- This is clearly better than just relying on black-box estimators
- Experimentally setting T to provide some overdesign

Bad news

- We did a great job with Pr[D[j]=1]
- We don’t know much about Pr[D[j]= prediction(D[j])]
- We haven’t thought much about ultimate source of unpredictability

Modeling the counts



- C[j] = # of transitions in span j
- We can measure this experimentally

Model: C[j] = V[j] + M

Assumptions:

- M doesn't change much in a small time
  - Might change over longer spans of time—seconds or minutes.
- V[j] has same distribution in every output

Model: C[j] = M + V[j] or C[j] ~= N(M,sigma)

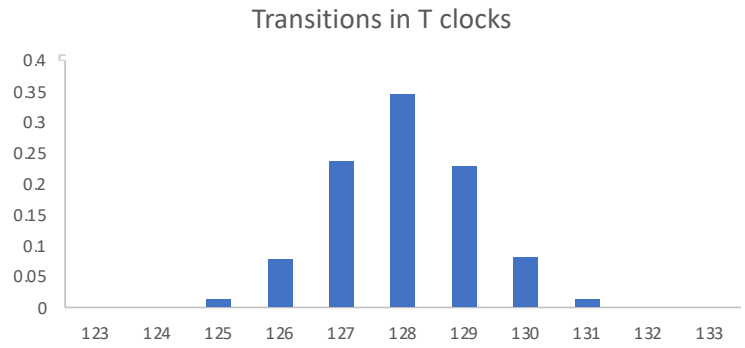
Measure C[j]

- Suppose when T is large, V[j] approximates a normal distribution
- Equivalently, C[j] ~ N(M,sigma) for some sigma

Test that C[j] plausibly follows same dist in all outputs

Test that transitions evenly divided in all clocks within an output

- So that doubling T really adds variability



- Now, we can get

P[D[j] = 0] = P[C[j] mod 2 = 0]

Good news/bad news

Good news

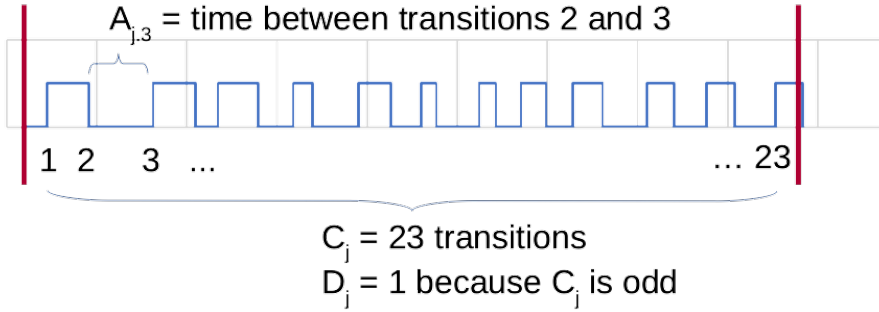
- We've delved a bit into internals of source
- We can support our entropy estimate this way
- We've checked many important assumptions
- Much higher confidence in estimate now

Bad news

- Haven't gotten down to actual source of noise

Modeling individual transition times

- A[j,k] is time taken for kth transition
  (0->1 or 1->0) in output j



Model:

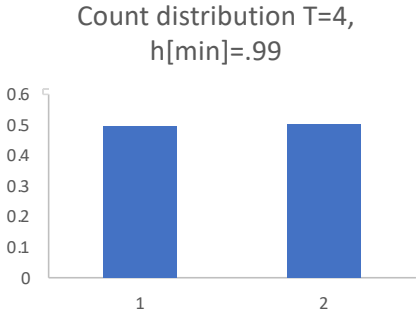
- A[j,k] ~ N(mu, sigma)

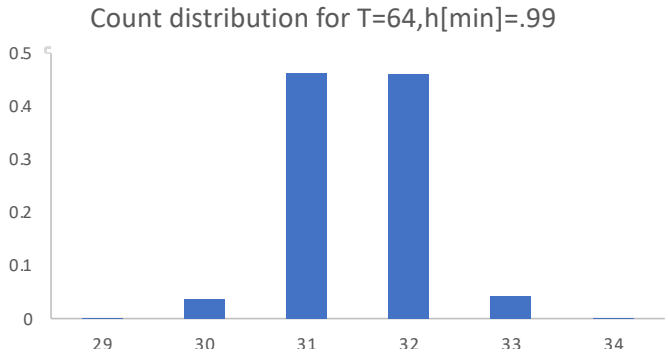
Measure parameters

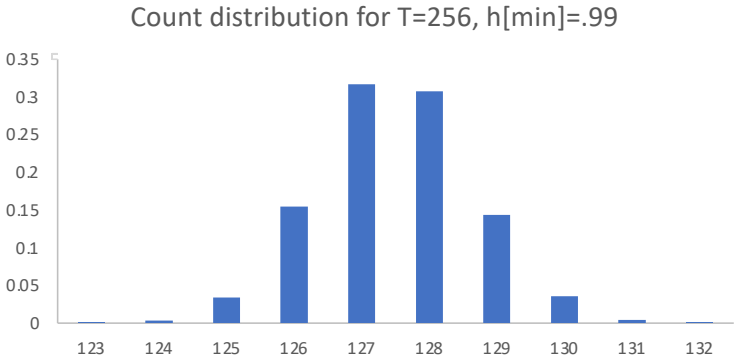
- Suppose we measure that mu~=40ns, sigma~=4 ns, one clock = 20 ns

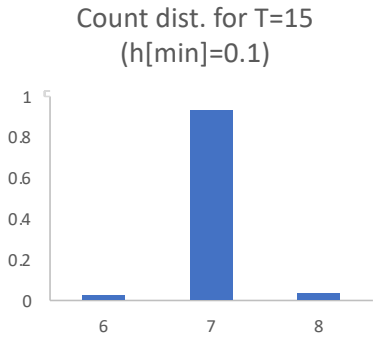
Use this model to predict Pr[D[j] = 0] and thus H[min]

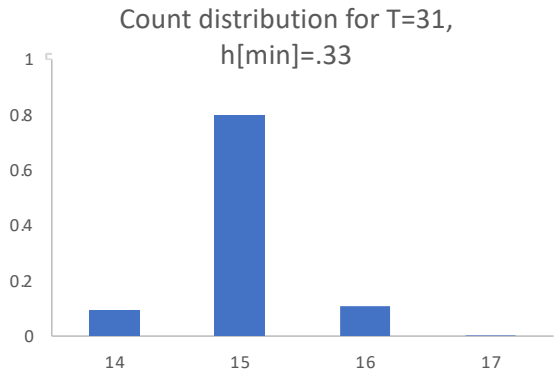
Some simulation results

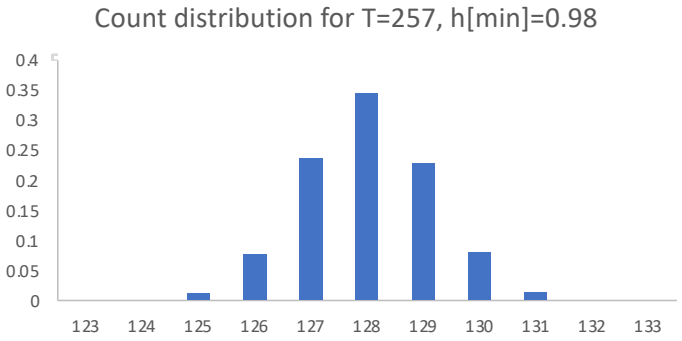




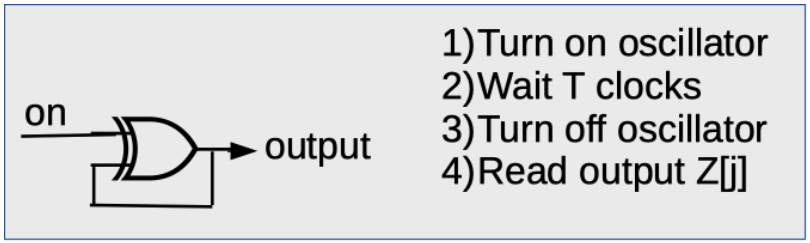








Where is the noise?



- Based on some measurements, calculations, or just literature, we might further estimate fraction of variability due to real noise

$$A[j,k] \sim N(mu, sigma\_\{noise\}) + F[j,k]$$

Where we assume F[j,k] is predictable to attacker

In this case, we can get a more conservative estimate of entropy

Model: clock = 20 ns, mu = 40 ns, sigma_{noise} = 2 ns

Entropy estimates for T=2,4,8,16,32,64 are all very similar (around 0.98)

Modeling Recap for on/off oscillator source

Going from least to most assurance

- Run black box estimators on outputs or on D[j] 🡐 Not enough!
- Experiment with T needed to make D[j] random, make T bigger
  - [Model: Each clock you get a weaker D, XOR together by making T bigger!]
- Measure and model transition counts, verify more of model
- Measure and model individual transition times
- Incorporate estimate of fraction of variability based on real noise

All these let us get an estimate for entropy/output

Black box estimation → Modeling outputs → modeling best possible predictions

Where else is the model used in 90B?

- Conditioning
- Health tests and failure conditions

Modeling and conditioning

- If a source uses a non-vetted conditioning function, then submission must justify why there won't be a bad interaction between source and conditioning function
- Model is how you do that.
- Example: Von Neumann unbiasing.
  - Draw pair (X,Y) from source
  - If X = Y, discard both and don't output anything
  - Else, output Y
- If source is iid and biased, result is unbiased
- If successive samples correlated, may make bias worse

Health tests and failure conditions

- Model should inform your health tests!

How can source fail?

- Total failure = model completely stops describing source
  - Something broke, oscillators locked to clock, etc.
- Model failure = assumed parameters of model are wrong
  - Causing entropy estimate to be inaccurate

- Health tests should be selected/designed based on what model says about source
  - How it can fail, what it will look like when estimated parameters are off

Modeling wrapup

- Noise sources MUST have some kind of model to justify entropy estimate
  - You need to bound prob that attacker can guess an output
  - Max_{all x} Pr[output = x | all attacker information]
- Black box tests are NOT enough
  - Necessary but not sufficient
- Future version of 90B will tighten requirements further
  - Look at AIS31 stochastic models for where we're trying to head
- Physical and non-physical models different
  - Non-physical sources are harder to get a good model
  - …mainly because they weren't designed to be a noise source

Questions about the noise source

- **How does the noise source work? (What's unpredictable about it?)**

  "The operation of the noise source shall be documented…"

- **Where does the unpredictability come from?**

  "…where the unpredictability comes from"

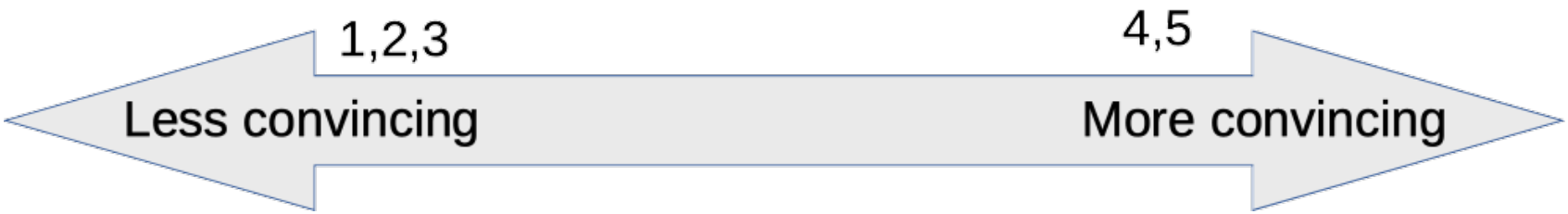
- **How much entropy / output is produced?**

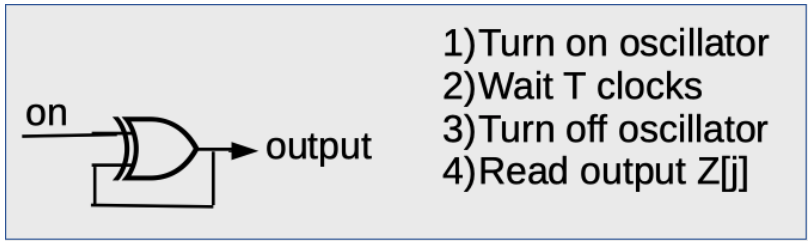
  "Documentation shall provide an explicit statement of the expected entropy provided…"

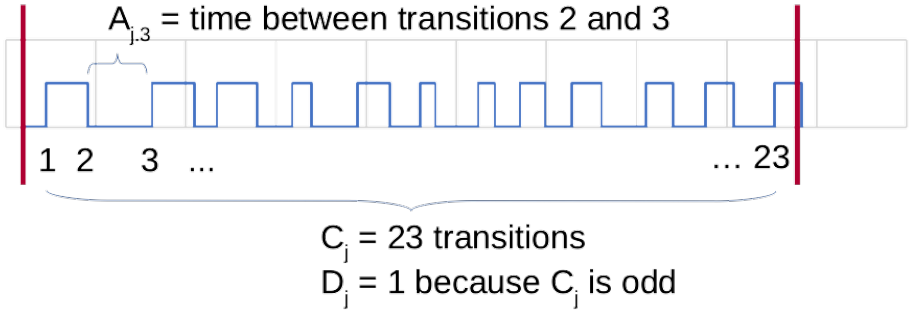
- **How do you know? (Justify the entropy estimate.)**

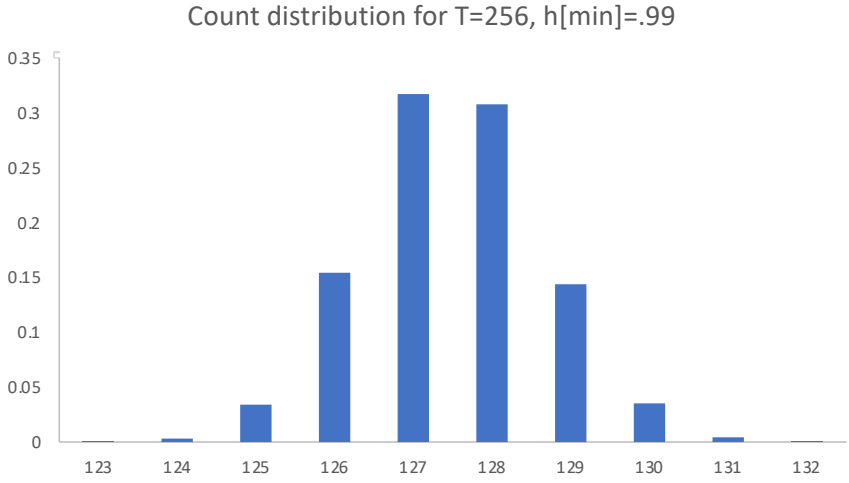
  "…provide a technical argument for why the noise source can support that entropy rate."

Questions?


1,2,3
4,5
Less convincing
More convincing


on
output
1)Turn on oscillator
2)Wait T clocks
3)Turn off oscillator
4)Read output Z[j]


A j.3 = time between transitions 2 and 3
1 2 3 ...
... 23
C j = 23 transitions
D j = 1 because C j is odd


Count distribution for T=256, h[min]=.99
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
123
124
125
126
127
128
129
130
131
132